日　本　国　特　許　庁
PATENT OFFICE
JAPANESE GOVERNMENT

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日
Date of Application: ２０００年１０月２３日

出願番号
Application Number: 特願２０００－３２２４０２

出願人
Applicant (s): 富士通株式会社

２０００年１２月１５日

特許庁長官
Commissioner,
Patent Office 及川耕造

出証番号　出証特２０００－３１０４７２１

【書類名】　特許願

【整理番号】　0052300

【提出日】　平成12年10月23日

【あて先】　特許庁長官殿

【国際特許分類】　G06F 09/06

【発明の名称】　データ構造解決ユニットを用いたプログラム自動生成方式

【請求項の数】　10

【発明者】

【住所又は居所】　神奈川県川崎市中原区上小田中４丁目１番１号　富士通株式会社内

【氏名】　橋本　惠二

【発明者】

【住所又は居所】　神奈川県川崎市中原区上小田中４丁目１番１号　富士通株式会社内

【氏名】　中條　有規

【発明者】

【住所又は居所】　神奈川県川崎市中原区上小田中４丁目１番１号　富士通株式会社内

【氏名】　浅井　景粋

【発明者】

【住所又は居所】　神奈川県川崎市中原区上小田中４丁目１番１号　富士通株式会社内

【氏名】　銀林　純

【発明者】

【住所又は居所】　神奈川県川崎市中原区上小田中４丁目１番１号　富士通株式会社内

【氏名】　吉田　裕之

【発明者】

【住所又は居所】　神奈川県川崎市中原区上小田中４丁目１番１号　富士通株式会社内

【氏名】　山本　里枝子

【発明者】

【住所又は居所】　神奈川県川崎市中原区上小田中４丁目１番１号　富士通株式会社内

【氏名】　濱本　惠美子

【特許出願人】

【識別番号】　000005223

【氏名又は名称】　富士通株式会社

【先の出願に基づく優先権主張】

【出願番号】　特願2000- 15296

【出願日】　平成12年 1月25日

【代理人】

【識別番号】　100103528

【弁理士】

【氏名又は名称】　原田　一男

【電話番号】　045-290-2761

【手数料の表示】

【予納台帳番号】　076762

【納付金額】　21,000円

【提出物件の目録】

【物件名】　明細書　1

【物件名】　図面　1

【物件名】　要約書　1

【包括委任状番号】　9909129

【プルーフの要否】　要

【書類名】　　明細書

【発明の名称】　　データ構造解決ユニットを用いたプログラム自動生成方式

【特許請求の範囲】

【請求項１】

所定の処理を行うプログラムを自動的に生成するプログラム自動生成装置であって、

前記所定の処理固有の設定を行うための解決ロジックを含み且つ予め対応付けられたデータ構造のための雛型プログラムを各々含む、複数のデータ構造解決ユニットと、

選択されたデータ構造に対応する前記データ構造解決ユニット内の前記雛型プログラムに含まれる前記解決ロジックの、前記所定の処理固有の設定に関する解決情報を取得し、当該解決ロジックの解決情報と前記雛型プログラムとを合成することにより、前記所定の処理を行うプログラムを生成する解決器と、

を有するプログラム自動生成装置。

【請求項２】

前記解決器が、

前記選択されたデータ構造に対応する前記データ構造解決ユニット内の前記雛型プログラムに含まれる前記解決ロジックを解析し、当該解決ロジックに対する解決情報の入力をユーザに促す手段

を有することを特徴とする請求項１記載のプログラム自動生成装置。

【請求項３】

前記選択されたデータ構造に対応する前記データ構造解決ユニットは、単純型データ構造に対応するデータ構造解決ユニット、伝票型データ構造に対応するデータ構造解決ユニット、階層型データ構造に対応するデータ構造解決ユニット、ツリー型データ構造に対応するデータ構造解決ユニット、在庫型データ構造に対応するデータ構造解決ユニット、時間帯予約型データ構造に対応するデータ構造解決ユニット、計画型データ構造に対応するデータ構造解決ユニット、座席予約型データ構造に対応するデータ構造解決ユニット、構成型データ構造に対応するデータ構造解決ユニット、明細主導伝票型データ構造に対応するデータ構造解決

ユニット、系図型データ構造に対応するデータ構造解決ユニット、及びマトリックス型データ構造に対応するデータ構造解決ユニットのうちいずれかのデータ構造解決ユニットであることを特徴とする請求項１記載のプログラム自動生成装置。

【請求項４】

前記データ構造解決ユニットが、

１又は複数のレコードタイプと、前記複数のレコードタイプが存在する場合には当該複数のレコードタイプ間のリンクとから構成されるデータ構造を規定し、当該所定のデータ構造についての前記所定の処理のための設定を行うための解決ロジックを含む第１の雛型プログラムと、

操作について前記所定の処理のための設定を行うための解決ロジックを含み、前記所定のデータ構造に対して実行される基本操作に対応する第２の雛型プログラムと、

を含むことを特徴とする請求項１記載のプログラム自動生成装置。

【請求項５】

所定の処理を行うプログラムを自動的に生成するプログラム自動生成プログラムを格納する記録媒体であって、

前記プログラム自動生成プログラムは、

前記所定の処理固有の設定を行うための解決ロジックを含み且つ予め対応付けられたデータ構造のための雛型プログラムを含むデータ構造解決ユニットにおいて、前記雛型プログラムに含まれる前記解決ロジックの、前記所定の処理固有の設定に関する解決情報を取得し、当該解決ロジックの解決情報と前記雛型プログラムとを合成することにより、前記所定の処理を行うプログラムを生成する解決プログラム、

を含む記録媒体。

【請求項６】

所定の仕様に従った第１プログラムを生成するために使用される第２プログラムを格納する記録媒体であって、

前記第２プログラムは、

１又は複数のレコードタイプと、前記複数のレコードタイプが存在する場合には当該複数のレコードタイプ間のリンクとから構成されるデータ構造を規定し、当該所定のデータ構造についての前記所定の仕様に従った設定を行うための解決ロジックを含む第１の雛型プログラムと、

操作について前記所定の仕様に従った設定を行うための解決ロジックを含み、前記所定のデータ構造に対して実行される基本操作に対応する第２の雛型プログラムと、

を含む記録媒体。

【請求項７】

前記第１の雛型プログラムが、１種類のヘッダ・レコードタイプと１種類の明細レコードタイプと１つの前記ヘッダ・レコードタイプに対し１又は複数の前記明細レコードタイプを結ぶリンクとを有するデータ構造を規定し、前記ヘッダレコード及び前記明細レコードの属性を与えるための解決ロジックを含み、

前記第２のプログラムが、前記データ構造に対して少なくとも伝票新規作成及び伝票検索を実行するための第２の雛型プログラムであって、前記ヘッダ・レコードの状態を操作との関連で定義するための解決ロジックと、前記所定の仕様に従った設定をレコードの属性、レコードの状態、若しくはレコードの属性及び状態の組合せで記述するための解決ロジックとが埋め込まれている、

ことを特徴とする請求項６記載の記録媒体。

【請求項８】

前記第１の雛型プログラムが、資源レコードタイプと、資源の予約に関する予約レコードタイプと、予約の時間の刻みに関する予約セル・レコードタイプと、１つの資源レコートタイプと１又は複数の予約レコードタイプとを結ぶリンクと、１つの予約レコードタイプと１又は複数の予約セル・レコードタイプを結ぶリンクとから構成されるデータ構造を規定し、前記データ構造内の各レコードの属性を与えるための解決ロジックを含み、

前記第２の雛型プログラムが、前記所定の仕様に従った設定を行うための解決ロジックを含み、前記データ構造に対して少なくとも予約登録、予約削除、予約更新及び予約検索を実行するためのものである、

ことを特徴とする請求項６記載の記録媒体。

【請求項９】

前記第１の雛型プログラムが、在庫レコードタイプと、在庫の引当を規定するための在庫引当明細レコードタイプと、入荷予定レコードタイプと、入荷予定に対する引当を規定するための入荷予定引当明細レコードタイプと、前記在庫レコードタイプと前記在庫引当明細レコードタイプとを結ぶリンクと、前記入荷予定レコードタイプと前記入荷予定引当明細レコードタイプとを結ぶリンクとを有するデータ構造を規定し、前記データ構造内の各レコードの属性を与える解決ロジックを含み、

前記第２のプログラムが、前記データ構造に含まれる各レコードに対し少なくとも追加操作、削除操作、及び検索操作を実行するための第２の雛型プログラムであって、前記所定の仕様に従った設定を行うための解決ロジックを含む、

ことを特徴とする請求項６記載の記録媒体。

【請求項１０】

前記第１の雛型プログラムが、資源レコードタイプと、資源のグループを規定するための資源グループ・レコードタイプと、資源グループを利用可能な機会を規定する機会レコード・タイプと、機会と資源の組み合わせを規定するためのオカーレンス・レコードタイプと、１又は複数のオカーレンスに関連する予約レコード・タイプとを含むデータ構造を規定し、前記データ構造内の各レコードの属性を与える解決ロジックを含み、

前記第２の雛型プログラムが、前記データ構造に含まれる各レコードに対し少なくとも生成操作、削除操作、及び検索操作を実行するための第２の雛型プログラムであって、前記所定の仕様に従った設定を行うための解決ロジックを含む、

ことを特徴とする請求項６記載の記録媒体。

【発明の詳細な説明】

【０００１】

【発明が属する技術分野】

本発明は、プログラムの作成を支援する技術に関し、より詳しくは、各データ構造に対応する雛型プログラムを用いたプログラムの自動生成技術に関する。

【０００２】

【従来の技術】

従来のプログラム作成支援システムでは、作成すべきプログラムに対する仕様を記述するための形式が用意され、それに従って記述された仕様からプログラムを生成するジェネレータ方式が一般的である。この方式では、（ａ）仕様記述の量を作成すべきプログラムの規模に比べてどの程度小さく出来るか、（ｂ）仕様の形式を生成すべきプログラムの構造よりもユーザに分かりやすくし、どの程度誤りが入り込みにくくできるか、という点が重要な評価対象となる。この（ａ）に注目すると、（ｉ）作成すべきプログラムを用途／処理フローに応じてパターン化し、パターン毎に雛型を用意して、固有の部分だけをユーザが記述する方法と、（ii）作成すべきプログラムよりも記述性が高い形式を仕様記述言語として提供する方法とがある。

【０００３】

【発明が解決しようとする課題】

（ｉ）の方法を採用する際の問題は、ユーザによる仕様の記述量を少なくするには雛型が相当量カバーする必要があるが、そのためには用途／処理フローに応じたパターンを生成すべきプログラムの多様性に応じて数多く用意しなければならない点にある。

【０００４】

本発明の目的は、より少ない雛型を用いて、多様なプログラムを生成できるようにするプログラム自動生成技術を提供することである。

【０００５】

【課題を解決するための手段】

本発明の第１の態様に係る、所定の処理を行うプログラムを自動的に生成するプログラム自動生成装置は、所定の処理固有の設定を行うための解決ロジックを含み且つ予め対応付けられたデータ構造のための雛型プログラムを各々含む、複数のデータ構造解決ユニットと、選択されたデータ構造に対応するデータ構造解決ユニット内の雛型プログラムに含まれる解決ロジックの、所定の処理固有の設定に関する解決情報を取得し、当該解決ロジックの解決情報と雛型プログラムと

を合成することにより、所定の処理を行うプログラムを生成する解決器とを有する。本発明ではデータ構造に対応して雛型プログラムが用意される。用途／処理フロー毎に雛型を用意するよりは、より少ない雛型の数にて多様なプログラムを生成できるようになる。

【０００６】

上で述べた解決器を、選択されたデータ構造に対応するデータ構造解決ユニット内の雛型プログラムに含まれる解決ロジックを解析し、当該解決ロジックに対する解決情報の入力をユーザに促す手段を有するような構成も可能である。これによりユーザはより簡単に解決ロジックに対する解決情報を入力できるようになる。よって、生成されるプログラムに誤りが入りにくくなる。

【０００７】

なお、上で述べたようなプログラム自動生成装置を、通常のコンピュータとプログラムの組み合せにて実装することができ、当該プログラムは、例えばフロッピーディスク、ＣＤ－ＲＯＭ、光磁気ディスク、半導体メモリ、ハードディスク等の記憶媒体又は記憶装置に格納される。また、処理途中の中間的なデータは、コンピュータのメインメモリなどの記憶装置に格納される。

【０００８】

本発明の第２の態様に係る、所定の仕様に従った第１プログラムを生成するために使用される第２プログラムは、１又は複数のレコードタイプと、複数のレコードタイプが存在する場合には当該複数のレコードタイプ間のリンクとから構成されるデータ構造を規定し、当該所定のデータ構造についての所定の仕様に従った設定を行うための解決ロジックを含む第１の雛型プログラムと、操作について所定の仕様に従った設定を行うための解決ロジックを含み、所定のデータ構造に対して実行される基本操作に対応する第２の雛型プログラムとを含む。第２プログラムは、上記のデータ構造解決ユニットである。

【０００９】

なお、データ構造解決ユニットは、後に述べる実施の形態では、単純型データ構造に対応するデータ構造解決ユニット、伝票型データ構造に対応するデータ構造解決ユニット、階層型データ構造に対応するデータ構造解決ユニット、ツリー

型データ構造に対応するデータ構造解決ユニット、在庫型データ構造に対応するデータ構造解決ユニット、時間帯予約型データ構造に対応するデータ構造解決ユニット、計画型データ構造に対応するデータ構造解決ユニット、座席予約型データ構造に対応するデータ構造解決ユニット、構成型データ構造に対応するデータ構造解決ユニット、明細主導伝票型データ構造に対応するデータ構造解決ユニット、系図型データ構造に対応するデータ構造解決ユニット、及びマトリックス型データ構造に対応するデータ構造解決ユニットのうちいずれかのデータ構造解決ユニットとである。

【００１０】

【発明の実施の形態】

本発明の一実施の形態に係るプログラム自動生成装置の概要を図１に示す。プログラム自動生成装置１は、データ構造解決ユニット３と、解決ロジック解析部７及び合成部９を含む解決器５と、解決情報入力画面１１とを有している。なお、データ構造解決ユニット３は、データ構造毎に複数用意されており（図１における３ａ，３ｂ，３ｃ）、ユーザが選択したデータ構造に対応するデータ構造解決ユニット３に対し解決器５が作用する。データ構造解決ユニット３は、生成すべきプログラム固有の設定を行うための解決ロジック３３を含む雛型プログラム３１を含む。解決器５の解決ロジック解析部７は、データ構造解決ユニット３の解決ロジック３３を解析し、解決情報をユーザに入力するように促す入力画面１１を生成する。ユーザは生成すべきプログラムの仕様に従って、解決ロジック３３に対する解決情報を入力画面１１に入力する。解決器５の合成部９は、データ構造解決ユニット３の雛型プログラム３１と、ユーザにより入力された、解決ロジック３３に対する解決情報を合成して、対象とするプログラム１３を生成する。

【００１１】

図２にデータ構造解決ユニット３の概要を示す。データ構造解決ユニット３のデータ構造部３２０は、本データ構造解決ユニット３に対応するデータ構造を規定する部分である。すなわち、１又は複数のレコードタイプ３２２及び３２４と、複数のレコードタイプが存在する場合には当該複数のレコードタイプの関連を

表すリンク３２６とによりデータ構造が規定される。このデータ構造部３２０は、データ構造を規定する他、データベース３９０とのインターフェースを有している。すなわち、レコードタイプ３２２及び３２４に格納されるデータをデータベースに出力する機能、データベースからデータを読み出し、レコードタイプ３２２及び３２４とレコードタイプ間の関連に従った構造に成形する機能とを有する。なお、データベース３９０はデータ構造解決ユニット３には含まれない。よって図２ではデータベース３９０を点線で描いている。

【００１２】

このデータ構造部３２０は実態的には雛型プログラム３８０である。雛型プログラム３８０は上で述べたように、当該データ構造解決ユニット３が取り扱うデータ構造を規定しており、データベース３９０とのインターフェースを有している。また、雛型プログラム３８０は解決ロジック３８２を含んでいる。この解決ロジック３８２は、当該データ構造解決ユニット３が取り扱うデータ構造についての設定を可能にするものである。例えば、レコードの属性や型を設定できるようにするものである。データ構造は既に決まっているので、データ構造部３２０に対応する雛型プログラム３８０内の解決ロジック３８２により設定できる部分は、後に述べる操作のための雛型プログラム内の解決ロジックに比べれば自由度は小さくなる。

【００１３】

操作基本部３１０は、データ構造部３２０により規定されたデータ構造に対して基本的な操作である基本操作３１２乃至３１６を実施する。基本操作３１２乃至３１６は、例えば、レコードの挿入、関連付け、更新、検索、削除等の操作であり、データ構造に対応して必要となる操作である。

【００１４】

この操作基本部３１０も実態的には雛型プログラム３５０乃至３７０である。図２では、雛型プログラム３５０は基本操作３１２に対応しており、雛型プログラム３６０は基本操作３１４に対応しており、雛型プログラム３７０は基本操作３１６に対応している。このように、基本操作３１２乃至３１６に対応する雛型プログラム３５０乃至３７０が用意される。雛型プログラム３５０乃至３７０は

、個々のレコードに含まれるデータ項目に対する操作を行うものであり、埋め込み可能なプログラムソースの形式で提供される。雛型プログラム３５０乃至３７０の埋め込み可能な部分が解決ロジック３５２である。解決ロジック３５２は、対象となるプログラム固有の業務ロジックを雛型プログラム３５０乃至３７０に埋め込む手段を提供するものである。実際的には、雛型プログラム３５０乃至３７０の中に、埋め込むべき場所、埋め込むべき情報、その形式がタグ付き言語にて用意されている。

【００１５】

１．伝票型データ構造の場合

本実施の形態をより理解し易いように、以下データ構造毎に説明する。例えば図３のような伝票操作画面１５を出力するようなプログラムを生成する場合を考える。伝票操作画面１５は購入品伝票を操作する画面であり、データ構造部３２０に係るデータと、操作基本部３１０に係る操作ボタン５００乃至５２０とを含む。データ構造部３２０に係るデータは、ヘッダ部４００に含まれる承認番号、承認区分及び扱い日付についてのテーブルと、明細部４１０に含まれる行番号及び購入品目のテーブルとを含む。操作基本部３１０に係る操作ボタンには、伝票登録操作を実施するための伝票登録ボタン５００、伝票更新操作を実施するための伝票更新ボタン５１０、伝票削除操作を実施するための伝票削除ボタン５２０等が含まれる。

【００１６】

図３のような伝票操作画面１５を出力するようなプログラムは、伝票型データ構造を使用している。よって、このようなプログラムを生成する際には、伝票型データ構造に対応するデータ構造解決ユニット３が選択されねばならない。伝票型データ構造では、ヘッダのまとまりで情報が取り扱われ、ヘッダは明細を管理する。伝票型データ構造に対応するデータ構造解決ユニット３のデータ構造部３２０は、例えば図４のように示すことが出来る。すなわち、キーと属性とを含むヘッダ部４００と、ヘッダ部４００からリンクされており、キーと属性とを含む明細部４１０とで表される。但し、明細部４１０は１つのヘッダ部４００に対してＮ件（Ｎは０以上の整数）リンクする。

【００１７】

また、データ構造解決ユニット３の基本操作部３１０は、例えば以下に示すような種類の基本操作を含む。なお括弧内の文字列はメソッド名である。

（１）ヘッダ検索（findHeaderByKey）

指定したヘッダのキーを持つ伝票のヘッダを検索する。

（２）明細複数件検索（findDetailsByKey）

指定したヘッダのキーを持つ伝票の明細を複数検索する。

（３）明細１件検索（findDetailByKey）

指定したヘッダのキーを持つ伝票の明細を行番号を指定して１件検索する。

（４）伝票検索（findSlipByKey）

指定したヘッダのキーを持つ伝票を検索する。

（５）条件によるヘッダ検索

ヘッダを条件指定で検索し、条件検索後に伝票１件分のヘッダ情報を取得する（nextHeader）。

（６）条件による伝票検索

伝票を条件指定で検索し、条件検索後に伝票１件分の全情報を取得する（nextSlip）。

（７）キーつきヘッダ新規作成（createHeaderWithKey1）

キーを設定して伝票のヘッダを新規作成し、結果を真偽で返す。

（８）キーつきヘッダ新規作成

キーを設定して伝票のヘッダを新規し（createHeaderWithKey2）、生成した伝票の情報を返す。

（９）キーなしヘッダ新規作成

キーを設定せずに伝票のヘッダを新規し（createHeaderWithoutKey）、生成した伝票の情報を返す。

（１０）キーつき伝票新規作成（createSlipWithKey1）

キーを設定して伝票を新規作成し、結果を真偽で返す。

（１１）キーつき伝票新規作成

キーを設定して伝票を新規作成し（createSlipWithKey2）、生成した伝票の情

報を返す。

（１２）キーなし伝票新規作成

キーを設定せずに伝票を新規作成し（createSlipWithoutKey）、生成した伝票の情報を返す。

（１３）明細追加（insertDetails1）

既存の伝票に明細を複数件追加し、結果を真偽で返す。

（１４）明細追加

既存の伝票に明細を複数件追加し（insertDetails2）、追加した伝票の情報を返す。

（１５）明細取消（cancelDetails）

既存の伝票の明細を業務的に取り消す。

（１６）ヘッダ更新

キーで指定したヘッダの内容を更新し、結果を真偽で返す。

（１７）ヘッダ更新

キーで指定したヘッダの内容を更新し、更新したヘッダの内容を返す。

（１８）明細更新

キーで指定した伝票の明細の内容を更新し、結果を真偽で返す。

（１９）明細更新

キーで指定した伝票の明細の内容を更新し、更新した伝票の内容を返す。

（２０）伝票更新

キーで指定した伝票の内容を更新し、結果を真偽で返す。

（２１）伝票削除（removeSlip）

指定したキーを持つヘッダを、そのヘッダが持っている全明細を含めて、物理的に削除する。

【００１８】

なお前段で説明した基本操作は、さらに細かい操作の集合である。例えば図５に示すように、伝票登録操作（前段の伝票新規作成）は、ヘッダ部４００に対して登録操作を実施し且つ明細部４１０に対して登録操作を実施する。伝票更新操作（前段の伝票更新）は、ヘッダ部４００に対して更新操作を実施し且つ明細部

４１０に対して更新操作を実施する。明細追加操作（前段の明細追加）は、明細部４１０に対して登録処理を実施する。

【００１９】

上でも述べたが基本操作部３１０の各基本操作に対応して解決ロジックを含む雛型プログラムが用意される。図６には伝票更新操作の雛型プログラムソースの例を示している。図６の例では、《》に囲まれている部分が解決ロジックであり、ユーザがこの部分に対する解決情報を入力することにより対象となるプログラムを生成できるようになる。図６の例では、操作名、伝票名、更新可能な状態名、更新後の状態名、固有チェック、及びエラーメッセージを、対象となるプログラムの仕様に従って設定する。図６は伝票更新操作の雛型プログラムである。

【００２０】

なお、《》に囲まれている情報だけでは複雑な記述はできないので、タグの意味内容を別途タグリストとして記述しておき、その中の情報を使用する場合もある。例えば、《》内にはタグリストの参照先を示しておき、例えばタグリストにおいてユーザに選択させるべき選択肢についてのデータを格納しておくことも可能である。

【００２１】

図１に示した解決器５の解決ロジック解析部７が上で述べたような伝票型データ構造に対応するデータ構造解決ユニット３の解決ロジック３３を解析すると、図７に示したような事項をユーザに入力するように促す。すなわち、解決ロジック解析部７は解決情報入力画面１１ａを出力する。入力画面１１ａは、データ構造部３２０の解決ロジックに関するデータ構造解決部６００及び６１０と、操作基本部３１０の解決ロジックに関する操作解決部６２０とに分けられる。

【００２２】

データ構造解決部６００は伝票型データ構造におけるヘッダ部４００についての設定を入力する部分である。図７では項目名「承認番号」が入力され、この承認番号はキーであり、型は整数（ｉｎｔ）型であるということが入力されている。すなわち、図４のヘッダ部４００のキーの部分には整数型の承認番号が格納される。また、文字列型（String）の項目名「承認区分」、及び日付型（date）の

項目名「扱い日付」という情報も入力されている。承認区分及び扱い日付は図４のヘッダ部４００の属性の部分に格納される。データ構造解決部６１０は伝票型データ構造における明細部４１０についての設定を入力する部分である。図７では整数型の項目名「承認番号」及び整数型の項目名「行番号」がキーとして入力されている。すなわち、図４の明細部４１０のキー部分に承認番号及び行番号が格納される。また文字列型の項目名「購入品目」も入力されている。この購入品目は図４の明細部４１０の属性部分に格納される。

【００２３】

操作解決部６２０は、図７では一部のみしか示されていない。ユーザが対象となるプログラムの仕様に従って、操作名として［課長承認］を入力する。そうすると、課長承認に対応する基本操作名を基本操作名リストから選択するようになっている。ここでは課長承認に対応する基本操作は伝票更新である。次に伝票更新操作に対応するロジック解決情報を入力する。解決ロジック解析部７は、伝票更新操作の雛型プログラムに埋め込まれた解決ロジックであるタグ付き言語の部分を解析して、更新可能状態名の具体的な入力（ここでは［承認申請中］）、更新後状態名の具体的な入力（ここでは［課長承認済］）、エラーメッセージの具体的な入力（ここでは［承認対象ではありません］）、固有チェックの具体的な入力（ここでは［承認区分＝”部長決済伝票”］。論理式でも記述できる）を求める。そして、ユーザは他の操作名を入力し、上で述べたのと同様な処理を繰り返す。図７の例では次に操作名として［伝票作成］を入力している。そして、伝票作成操作に対応する基本操作が伝票登録と選択される。

【００２４】

なお図７のように一画面で全ての解決情報を入力するようにするとユーザには煩雑になり、入力漏れや、入力過ちを生ずる可能性がある。よって、図８乃至図１２のように順番に必要な情報を入力／選択させるようにしてもよい。

【００２５】

図８は本プログラム自動生成装置１を使用してプログラムの自動生成を行う際に最初に行うべきデータ構造の選択のための型選択画面１１ｂである。型選択画面１１ｂは、単純型データ構造を選択するためのボタンと、伝票型データ構造を

選択するためのボタンと、階層型データ構造を選択するためのボタンと、ツリー型データ構造を選択するためのボタンと、在庫型データ構造を選択するためのボタンと、時間帯予約型データ構造を選択するためのボタンと、計画型データ構造を選択するためのボタンと、座席予約型データ構造を選択するためのボタンと、構成型データ構造を選択するためのボタンと、明細主導の伝票型データ構造を選択するためのボタンと、系図型データ構造を選択するためのボタンと、マトリックス型データ構造を選択するためのボタンと、本プログラム自動生成装置が出力する最初の画面に戻るためのボタン（戻る）と、選択したデータ構造に関する解決情報を入力する処理に移行するためのボタン（次へ）と、選択をキャンセルするためのボタンとが含まれる。型選択画面１１ｂにて選択されたデータ構造に対応するデータ構造解決ユニット３が以下の処理を行うために用意される。

【００２６】

本実施例では上で述べたように１２種類のデータ構造から選択できるようになっているが、これ以外のデータ構造を追加して、より多くの種類のデータ構造からプログラム生成のためのデータ構造を選択するようにすることも可能である。また、上で述べた１２種類のデータ構造のうち２以上の任意のデータ構造のみを選択可能として型選択画面１１ｂに提示するような態様も可能である。

【００２７】

ここでは伝票型データ構造が選択され、「次へ」のボタンが押されたものとする。そうすると、伝票型データ構造に対応するデータ構造解決ユニット３が取り出され、解決ロジック解析部７は、データ構造解決ユニット３に含まれる解決ロジック３３を解析し、まず図９に示す伝票型解決情報入力画面その１（１１ｃ）を出力する。この伝票型解決情報入力画面その１（１１ｃ）では、ユーザに伝票名を入力するよう促す。本例では購入品伝票を処理するプログラムを生成するために、ユーザは「購入品伝票」と入力し、「次へ」というボタンを押す。なお、データ構造を選択し直す場合には、「戻る」のボタンを押し、伝票名の入力をキャンセルする場合には「キャンセル」のボタンを押す。

【００２８】

図９で「次へ」のボタンが押されると、伝票型解決情報入力画面その２（１１

ｄ）が解決ロジック解析部７から出力される。解決ロジック解析部７はデータ構造部３２０の雛型プログラム３８０に埋め込まれた解決ロジック３８２を解析して、データ構造解決部６００及び６１０の入力用テーブルを出力する。データ構造解決部６００では、ユーザにヘッダ部４００の解決情報を入力するよう促す。ここでは、項目名、キーか否か、型を入力する。すなわち、雛型プログラム３８０にはヘッダ・レコードに属性を与えるための解決ロジック３８２が含まれている。本例では、キーであり、整数型の項目名「承認番号」と、文字列型の項目名「承認区分」と、日付型の項目名「扱い日付」が入力されている。データ構造解決部６１０では、ユーザに明細部４１０の解決情報を入力するよう促す。ここでは、項目名、キーか否か、型を入力する。すなわち、雛型プログラム３８０には明細レコードに属性を与えるための解決ロジックが含まれている。本例では、キーであり、整数型の項目名「承認番号」と、キーであり、整数型の項目名「行番号」と、文字列型の項目名「購入品目」が入力されている。

【００２９】

ユーザは生成すべきプログラムの仕様に従った解決情報の入力が終われば、次の解決情報の入力のため「次へ」ボタンを押す。なお、伝票名の入力をやり直す場合には「戻る」ボタンを押す。また、入力をキャンセルする場合には「キャンセル」ボタンを押す。

【００３０】

図１０で「次へ」ボタンが押されると、伝票型解決情報入力画面その３（１１ｅ）が解決ロジック解析部７から出力される。解決ロジック解析部７はどの基本操作に対する設定を行うかの選択をユーザに促す。伝票解決情報入力画面その３（１１ｅ）では、操作名を入力し、この操作名に対応する基本操作名を選択する。基本操作名の選択は、基本操作部３１０に含まれる基本操作の名前に関する基本操作名リストを使用する。図１１では基本操作名リストをコンボボックスで表示するようになっている。図１１の例では操作名には課長承認が入力され、それに対応する基本操作には伝票更新が選択される。なお、コンボボックスの表示にて隠れているが、伝票型解決情報入力画面その３（１１ｅ）の入力が終了すると、「次へ」ボタンを押す。一方、伝票型解決情報入力画面その２（１１ｄ）に戻

る場合には「戻る」ボタンを押す。入力をキャンセルする場合には「キャンセル」ボタンを押す。

【００３１】

図１１で「次へ」のボタンが押されると、解決ロジック解析部７から伝票解決情報入力画面その４（１１ｆ）が出力される。この伝票解決情報入力画面その４（１１ｆ）は、操作解決部６２０の入力をユーザに促すものである。解決ロジック解析部７は、選択された基本操作に対応する雛型プログラムを解析し、埋め込まれている解決ロジックに対する解決情報の入力を求める。本例では、更新可能状態名の入力、更新後状態名の入力、エラーメッセージの入力、固有チェックの入力が可能となっており、更新可能状態名には「承認申請中」、更新後状態名には「承認申請済」、エラーメッセージには「承認対象ではありません」、固有チェックには「承認区分＝”部長決済伝票”」が入力されている。このように、基本操作に対応する本雛型プログラムには、ヘッダ・レコードの状態を操作との関連で定義するための解決ロジックと、プログラムの仕様に従った設定をレコードの属性、レコードの状態、若しくはレコードの属性及び状態の組合せで記述するための解決ロジックが埋め込まれている。

【００３２】

伝票型解決情報入力画面その４（１１ｆ）の入力が終了すると、生成ボタンを押す。これにより、基本操作の伝票更新操作に係るプログラムが合成部９により生成される。続いて、生成すべきプログラムの仕様において必要とされる他の操作に対する解決情報の入力が行われる。なお、前の入力画面に戻る場合には「戻る」ボタンを押し、入力をキャンセルする場合には「キャンセル」ボタンを押す。

【００３３】

以上のように最初にデータ構造がユーザにより選択され、解決ロジック解析部７が選択されたデータ構造に対応するデータ構造解決ユニット３の解決ロジック３３を解析し且つ解決情報入力画面１１を出力し、それに対応して解決情報がユーザにより入力され、合成部９が解決情報と雛型プログラム３１を合成することによりプログラムを生成する。

【００３４】

ここで解決器５の処理フローを図１３乃至図１５まとめておく。

【００３５】

ユーザによりデータ構造が選択されて処理が開始されると（図１３：ステップＳ１１）、選択されたデータ構造に対応するデータ構造解決ユニット３の解析が解決ロジック解析部７により行われる。最初に、データ構造解決ユニット３のデータ構造部３２０の解析が行われる（ステップＳ１３）。雛型プログラム３８０の解決ロジック３８２が解析される。解析の結果は後で用いるので、一旦記憶装置に格納される。

【００３６】

次に、操作基本部３１０の解析を行う（ステップＳ１５）。ここでは、操作基本部３１０の基本操作毎の雛型プログラム３５０乃至３７０を読み込み、基本操作の名前のリストである基本操作名リストを作成する。そして、読み込んだ雛型プログラム３５０乃至３７０の解析を実施する（ステップＳ１７）。雛型プログラムの解析については後に図１４を用いて説明する。

【００３７】

ステップＳ１７までにデータ構造解決ユニット３の解析を終了すると、解決ロジック解析部７は解決ロジックに対する解決情報を入力するための解決情報入力画面の生成を行い、ユーザに提示する（ステップＳ１９）。そして、ユーザから解決情報を取得する。このステップＳ１９の処理についても後に図１５を用いて説明する。合成部９は、取得した解決情報と雛型プログラム３１を合成し（ステップＳ２１）、対象プログラムを生成する（ステップＳ２３）。なお、雛型プログラム３１に埋め込まれた解決ロジックの部分を取得した解決情報で埋めれば、対象となるプログラムが生成できる。

【００３８】

では、図１４を用いて雛型プログラムの解析処理を示す。図１４の処理は伝票型データ構造の場合の解析処理である。まず、伝票型データ構造に対応する雛型プログラムに埋め込まれた解析ロジックである伝票名タグの存在を確認する（ステップＳ３３）。伝票型データ構造に対応する雛型プログラムには必ず伝票名タ

グが含まれる。よって、もしステップＳ３３で伝票名タグが見つからない場合には、当該雛型プログラムは伝票型データ構造に対応する雛型プログラムとしては適当でないということになる。存在が確認できなければ、エラーメッセージを出力するようにしてもよい。

【００３９】

次に操作名タグの存在を確認する（ステップＳ３５）。この操作名タグも雛型プログラムには必ず設けられた解決ロジックであるから、操作名タグが見つからない場合にはエラーメッセージを出力するようにしてもよい。そして、雛型プログラムを走査してその他のタグを検索し、タグ内容の確認及び記憶装置への格納を行う（ステップＳ３９）。雛型プログラムの解析結果は、一旦記憶装置に格納され、次の解決情報入力画面生成処理に用いられる。

【００４０】

次に、解決情報入力画面の生成及び解決情報の取得処理について図１５を用いて説明する。図１５の処理は伝票型データ構造の場合の処理である。まず、伝票名入力画面を作成し、ユーザに対して出力する。ユーザからは伝票名を取得する（ステップＳ４３）。例えば図９に示した伝票型解決情報入力画面その１（１１ｃ）を出力し、ユーザから伝票名を取得する。次に、データ構造入力画面を作成し、ユーザに対して出力する。ユーザからはデータ構造についての設定内容を取得する（ステップＳ４４）。例えば図１０に示した伝票型解決情報入力画面その２（１１ｄ）を出力し、ユーザからヘッダ部４００及び明細部４１０に関する情報を取得する。

【００４１】

そして、基本操作名リストからの選択部分を含む操作内容入力画面を作成し、ユーザに対して出力する。ユーザからは操作内容を取得する（ステップＳ４５）。例えば図１１に示した伝票型解決情報入力画面その３（１１ｅ）を出力し、ユーザから操作名及び対応する基本操作名を取得する。最後に、その他のタグの解決情報入力画面を作成し、ユーザに対して出力する。ユーザからはその他の解決情報を取得する（ステップＳ４７）。例えば図１２に示した伝票型解決情報入力画面その４（１１ｆ）を出力し、ユーザからは対象となるプログラムの固有の設

定に関する解決情報を取得する。

【００４２】

次に伝票型データ構造を使用した他のプログラムを生成する場合に入力すべき解決情報の例を示す。図１６は出荷伝票を処理するプログラムを作成する際に入力すべき解決情報を入力するための画面例１１ｇを示す。図７に示したのと同じように、伝票型データ構造の場合には、データ構造解決部６４０がヘッダ部４００に対する設定を入力する部分であり、データ構造解決部６５０が明細部４１０に対する設定を入力する部分である。ヘッダ部４００には、項目名、キーか否か、データの型が入力される。図１６の例では、項目名「出荷番号」はキーであり、整数型のデータであることが入力されている。項目名「出荷区分」は文字列型のデータであることが入力されている。項目名「扱い日付」が日付型のデータであることが入力されている。

【００４３】

明細部４１０には、項目名、キーか否か、データの型が入力される。図１６の例では、項目名「出荷番号」はキーであり、整数型のデータであることが入力されている。項目名「行番号」はキーであり、整数型のデータであることが入力されている。項目名「商品コード」は文字列型のデータであることが入力されている。

【００４４】

操作解決部６６０では、操作名を入力し、その後対応する基本操作の雛型プログラムの解決ロジックに解決情報を埋め込むための処理が行われる。図１６の例では、まず［出荷確定］が操作名として入力される。この出荷確定に対応する基本操作は、基本操作名リストから選択される。ここでは伝票更新が対応する基本操作として選択されている。そして、この伝票更新操作の雛型プログラムの解決ロジックに対応する解決情報が入力される。ここでは、更新可能状態名の入力が求められ、ユーザにより［出荷依頼済］が入力されている。また更新後状態名の入力が求められ、ユーザにより［出荷確定済］が入力されている。またエラーメッセージの入力が求められ、ユーザにより［出荷依頼が行われておりません。］が入力されている。さらに固有チェックの入力が求められ、ユーザにより［出荷

伝票が当月度出荷対象か。］が入力されている。出荷確定操作に対する入力が終了すると、他の操作に対する解決情報の入力が行われる。図１６の例では次に操作名［出荷依頼］が入力され、対応する基本操作として伝票登録が選択されたことが示されている。このような処理が対象となるプログラムの仕様に沿って必要な操作を規定し終えるまで繰り返される。

【００４５】

２．時間帯予約型データ構造の場合

次に時間帯予約型データ構造について説明する。時間帯予約型データ構造を選択した場合に生成されるプログラムが出力する画面の一例を図１７に示す。このプログラムは会議室予約プログラムであり、会議室予約画面を出力する。表８００は、データ構造解決ユニット３のデータ構造部３２０に対応するデータを表示する部分である。ここでは応接Ａ、応接Ｂ、談話室という資源８８０に対する予約を行うことができる。図１７では６月７日と、６月８日と、６月９日とに関する予約処理を行う場面を示している。日付毎に、予約する時間を分かりやすくするために９時から１７時までの時間軸が示されている。各資源の行には、予約が既に入っている時間帯を示す帯が示されている。一方、予約登録のためのボタン８１０、予約取り消しのためのボタン８２０、キャンセルのためのボタン８３０は、データ構造解決ユニット３の操作基本部３１０に対応する操作を実行させるためのボタン群を示している。

【００４６】

図１７の応接Ａの６月９日の予約状況を拡大したのが拡大表示部８４０である。予約を行う時間軸の部分をグリッド８７０と呼ぶ。予約の時間単位を予約セル８５０とし、予約セル８５０の集合として予約８６０が定義される。

【００４７】

では、図１７のような画面を出力するためのプログラムの元となる時間帯予約型データ構造を図１８に示す。時間帯予約型データ構造は、資源部４３０と、Ｍ個（Ｍは０以上の整数）の予約部４４０と、１つの予約部４４０に対してＮ個（Ｎは０以上の整数）の予約セル部４５０と、資源部４３０と予約部４４０とを結ぶリンクと、予約部４４０と予約セル部４５０とを結ぶリンクとから構成される

。資源部４３０は図１７における資源８８０のためのデータを格納する部分である。予約部４４０は図１７における予約８６０のためのデータを格納する部分である。予約セル部４５０は図１７における予約セル８５０のためのデータを格納する部分である。資源部４３０、予約部４４０、予約セル部４５０にはそれぞれキーと属性が格納される。

【００４８】

また、データ構造解決ユニット３の基本操作部３１０は、例えば以下に示すような種類の基本操作を含む。なお括弧内の文字列はメソッド名である。

（１）予約登録（キーあり）（createReservationWithKey1）

入力情報に従い、予約と予約セルを創成（既に創成されている場合はキャンセル待ちとして登録）する。

（２）予約登録（キーあり）

入力情報に従い、予約と予約セルを創成（既に創成されている場合はキャンセル待ちとして登録）し（createReservationWithKey2）、創生した予約エンティティの情報を返す。

（３）予約登録（キーなし）

入力情報に従い、予約と予約セルを創成（既に創成されている場合はキャンセル待ちとして登録）し（createReservationWithoutKey）、創生した予約エンティティの情報を返す。なお、プライマリキーは採番クラスから取得して使用する。

（４）予約取消（cancelReservation）

入力情報の予約キーの予約を取り消す（物理削除）。関連する予約セルに対しては自予約の状態により以下の処理を行う。（ａ）自予約が「キャンセル待ち」であれば、その指定を解除する。（ｂ）自予約が「正予約」であり、キャンセル待ちが指定されていれば、キャンセル待ちを正予約に昇格させキャンセル待ちを解除する。（ｃ）自予約が「正予約」であり、キャンセル待ちが指定されていなければ、予約セルを削除する。

（５）正予約可否確認（isVacant）

入力情報の資源と日付、時間帯について対応する予約セルを確認し、予約セル

が存在しない場合に正予約可（true)を返す。

（６）予約不可確認（isOccupied)

入力情報の資源と日付、時間帯について対応する予約セルを確認し、キャンセル待ち指定済みが１つでもある場合は予約不可（true）を返す。

（７）予約状況確認（isReserved)

入力された予約キーの予約に対し、関連する全ての予約セルの状況を確認し、全てが「正予約」であれば予約成立（true）を返す。

（８）予約期間変更（changeReservedPeriod)

入力された予約キーの予約を取得し、指定された期間に変更する。変更後の期間が予約可能であれば変更を行うが、予約不可の場合は予約不成立（負の値）を返す。変更後の予約状況により、予約成立は正の値、キャンセル待ち状態なら０を返す。

（９）予約資源変更（changeReservedResource)

入力された予約キーの予約を取得し、指定された資源に変更する。指定された資源の変更後の期間が予約可能であれば変更を行うが、予約不可の場合は予約不成立（負の値）を返す。変更後の予約状況により、予約成立は正の値、キャンセル待ち状態なら０を返す。

（１０）予約更新

入力情報に従い、予約を更新する。結果は真偽で返される。

（１１）連鎖昇格予約キーリスト取得（getChainElevatedReservation)

予約取消、予約期間変更、予約資源変更の操作で、既存の正予約が取り消されたことによってキャンセル待ち予約が昇格した場合にその予約キーリストを指定されたメッセージキャリアに格納する。

（１２）予約キー検索（findReservationByKey)

指定された予約キーの予約を取得し、その情報を出力用メッセージキャリアに格納する。

（１３）期間指定予約検索

指定された日付（期間）に含まれる予約を検索し、存在するかどうかを返し（findReservationByPeriod）、予約の複数検索の結果を指定されたメッセージキ

ヤリアに格納して返す（nextReservation）。

（１４）資源指定予約検索

指定された資源キーの予約を検索し、存在するかどうかを返し（findReservationByResource）。予約の複数検索の結果を指定されたメッセージキャリアに格納して返す（nextReservation）。

（１５）資源期間指定予約検索

指定された資源キーと日付（期間）を条件に予約を検索し、存在するかどうかを返し（findReservationByResourceAndPeriod）、予約の複数検索の結果を指定されたメッセージキャリアに格納して返す（nextReservation）。

（１６）予約セルキー検索（findReservedCellByKey）

指定された予約セルキーの予約セルを取得し、指定されたメッセージキャリアに格納する。

（１７）キャンセル待ちキー指定予約検索（findReservationByCancelWaitingKey）

指定された予約キーがキャンセル待ちとなっている予約セルを検索し、その正予約キーのリストを出力用のメッセージキャリアに格納する。

（１８）予約キー指定キャンセル待ち検索（findCancelWaitingByReservationKey）

指定された予約キーが正予約となっている予約セルを検索し、そのキャンセル待ち予約キーのリストを出力用のメッセージキャリアに格納する。

（１９）資源追加（キーあり）

入力情報に従って資源を追加作成する（返却値は真偽）（insertResourceWithKey）。そして、追加された資源の情報を返す。

（２０）資源追加（キーなし）

入力情報に従って資源を追加作成する（insertResourceWithoutKey）。プライマリキーは採番クラスより取得する（返却値は真偽）。追加された資源の情報を返す。

（２１）資源削除（removeResource）

指定された資源キーの資源を削除（物理削除）する。

（２２）資源キー検索（findResourceByKey）

指定された資源キーの資源を取得し、情報を出力用メッセージキャリアに格納する。

（２３）資源条件検索

指定された条件に合う資源を検索し、存在するかどうかを返す。そして、資源の複数検索の結果を指定されたメッセージキャリアに格納して返す（nextResource）。

【００４９】

なお前段で説明した基本操作は、さらに細かい操作の集合である。例えば図１９に示すように、予約登録操作（前段の予約登録）は、資源部４３０に対しては登録処理を実施し、予約部４４０に対しては登録処理を実施し、予約セル部４５０に対しては登録処理を実施する。予約取消操作（前段の予約取消）は、資源部４３０に対しては削除処理を実施し、予約部４４０に対しては削除処理を実施し、予約セル部４５０に対しては削除処理を実施する。空状況確認操作（前段の資源指定予約検索等））は、予約部４４０に対して検索処理を実施する。予約期間変更操作（前段の予約期間変更）は、予約部４４０に対しては更新処理を実施し、予約セル部４５０に対しては削除又は登録処理を実施する。予約資源変更操作（前段の予約資源変更）は、資源部４３０に対しては検索処理を実施し、予約部４４０に対しては更新処理を実施し、予約セル部４５０に対しては削除又は登録処理を実施する。予約条件検索操作（前段の資源指定予約検索等）は、予約部４４０に対して検索処理を実施する。

【００５０】

時間帯予約型データ構造に対してこのようなデータ構造解決ユニット３が用意されており、解決ロジック解析部７はデータ構造解決ユニット３の解決ロジックを解析して、例えば図２０に示すような解決情報入力画面１１ｉを作成してユーザに出力する。

【００５１】

図７及び図１６に示したように、データ構造解決ユニット３のデータ構造部３２０に対する解決情報を入力するためのデータ構造解決部９００乃至９２０と、

データ構造解決ユニット３の操作基本部３１０に対応する雛型プログラム３８０に埋め込まれた解決ロジック３８２に対する解決情報を入力するための操作解決部９３０とが含まれる。

【００５２】

データ構造解決部９００は、時間帯予約型データ構造における資源部４３０の解決情報を入力する部分である。データ構造解決部９００には資源名に対する入力部が設けられており、ここでは［会議室］が入力されている。また、項目名、キーか否か、データの型を入力するようになっている。すなわち、雛型プログラム３８０には資源レコードに属性を与えるための解決ロジック３８２が含まれている。図２０の例では、項目名「会議室コード」が入力されており、この会議室コードはキーであり、文字列型のデータであることが入力されている。項目名「会議室名」が入力されており、会議室名は文字列型のデータであることが入力されている。項目名「収容人数」が入力されており、収容人数は整数型のデータであることが入力されている。項目名「ＴＶ会議可」が入力されており、ＴＶ会議可は論理型（boolean）のデータであることが入力されている。

【００５３】

データ構造解決部９１０は、時間帯予約型データ構造における予約部４４０の解決情報を入力する部分である。ここには、項目名、キーか否か、データの型を入力するようになっている。すなわち、雛型プログラム３８０には予約レコードに属性を与えるための解決ロジック３８２が含まれている。図２０の例では、項目名「予約番号」が入力されており、この予約番号はキーであり且つ整数型のデータであることが入力されている。項目名「会議室コード」が入力されており、この会議室コードはキーであり且つ文字列型のデータであることが入力されている。項目名「開始日付」が入力されており、日付型のデータであることが入力されている。項目名「開始グリッド番号（Ｎｏ）」が入力されており、この開始グリッド番号が整数型のデータであることが入力されている。

【００５４】

データ構造解決部９２０は、時間帯予約型データ構造における予約セル部４５０の解決情報を入力する部分である。すなわち、雛型プログラム３８０には予約

セル・レコードに属性を与えるための解決ロジック３８２が含まれている。ここでは、グリッド番号の最大値と、グリッド単位名が入力されるようになっている。図２０の例ではグリッド番号の最大値は２１であり、グリッド単位名が「時」であることが入力されている。

【００５５】

操作解決部９３０は、データ構造解決ユニット３の操作基本部３１０に対応する雛型プログラム３５０乃至３７０に埋め込まれた解決ロジック３５２に対する解決情報を入力する部分である。まず、操作名を最初に入力するようにユーザに求める。図２０では操作名は［会議室予約］と入力されている。この会議室予約に対応する基本操作を次に選択する。この選択には基本操作名リストが用いられる。図２０では会議室予約に対応する基本操作は予約登録と選択されている。この基本操作「予約登録」に対応する雛型プログラムが読み出され、埋め込まれた解決ロジックに代わって入力されるべき解決情報の入力が求められる。図２０では、予約可能日チェックという項目に対して［営業日チェック（予約日付）＝”ＯＫ”］が入力されている。エラーメッセージという項目に対して［予約できませんでした］が入力されている。固有チェックという項目に対して［所属チェック（所属コード）＝”ＯＫ”］が入力されている。予約可能日チェックと固有チェックは、チェックの条件を論理式で記述可能である。

【００５６】

以上解決情報を入力すれば、基本操作である予約登録についてのプログラムを生成することができる。対象とすべきプログラムの仕様に従って必要とされる基本操作についての解決情報を続いて入力するようにユーザに促す。図２０の例では、次に操作名［会議室変更］操作についての解決情報を入力するようになっている。この会議室変更に対応する基本操作は予約資源変更操作と選択されている。

【００５７】

３．単純型データ構造の場合

単純型データ構造は、図４に示された伝票型データ構造のようにリンクを有しておらず、例えばヘッダ部４００のみで存在するようなデータ構造である（図２

１参照）。例えば、「職制」のみを管理したり、一商品毎に別伝票を起こすような場合に適用できる。データ構造部３２０は、１つのレコードタイプを規定しており、１つのレコードタイプの属性を与えるための解決ロジックを含む雛型プログラムが対応する。なお、レコードタイプには、生成、削除、キー検索を実行する処理部が設けられる。

【００５８】

また、単純型データ構造の操作基本部３１０には、以下に示すような基本操作が含まれる。なお括弧内の文字列はメソッド名である。

（１）レコードの検索（findSimpleEntityByKey）

指定されたキーを持つエンティティ（＝レコードタイプ。以下同じ。）のインスタンスを検索する（返却値はエンティティの情報）。返却値はエンティティ及びその状態の情報の場合もある。

（２）レコードの追加（createSimpleEntityWithKey1）

指定されたキーを持つエンティティのインスタンスを生成する（返却値は真偽）。

（３）レコードの追加（createSimpleEntityWithKey2）

指定されたキーを持つエンティティのインスタンスを生成する（返却値はエンティティの情報）。

（４）レコードの追加（キーなし）（createSimpleEntityWithoutKey）

エンティティのインスタンスを生成する（返却値はエンティティの情報）。

（５）レコードの削除（removeEntity）

指定されたキーを持つエンティティのインスタンスを削除する。

（６）レコードの状態参照（getState）

指定されたキーを持つエンティティのインスタンスから状態を取得する。

（７）レコードの条件検索

条件に適合するエンティティのインスタンスの集合を準備し、その準備ができたかどうかを返す。そして、次のエンティティの情報を取得する（nextSimpleEntity）。

（８）レコードの更新

指定されたキーを持つエンティティのインスタンスの情報を更新する（返却値は真偽）。

（９）レコードの更新

指定されたキーを持つエンティティのインスタンスの情報を更新する（返却値は更新したエンティティの情報）。

このように単純型データ構造に対する基本操作に対応する各雛型プログラムには、対象となるプログラム固有の設定を行うための解決ロジックが埋め込まれている。

【００５９】

４．マトリックス型データ構造の場合

次にマトリックス型データ構造に係るデータ構造解決ユニット３について説明する。このマトリックス型データ構造は、図２２に示すように行部４６２と、列部４６０と、行及び列の交点のレコードを示すセル部４６４と、各列のタイプを規定し且つ各セル部４６４に対して割り当て可能な属性を表す列タイプ部４６６と、行部４６２とセル部４６４とのリンクと、列部４６０とセル部４６４とのリンクと、列部４６０と列タイプ部４６６とのリンクと、列タイプ部４６６とセル部４６４とのリンクとを含む。なお、１つの行部４６２に対してＮ個（Ｎは０以上の整数）のセル部４６４が存在する。また、１つの列部４６０に対してＭ個（Ｍは０以上の整数）のセル部４６４が存在する。また、１つの列部４６０に対してＫ個（Ｋは０以上の整数）の列タイプ部４６６が存在する。また、１つの列タイプ部４６６に対してＬ個（Ｌは０以上の整数）のセル部４６４が存在する。なお、データ構造部３２０には、行部４６２と、列部４６０と、列タイプ部４６６とに属性を与えるための解決ロジックが含まれる。また、行部４６２、列部４６０、セル部４６４及び列タイプ部４６６の各々には、生成、削除、検索等を実行する処理部が設けられている。

【００６０】

マトリックス型データ構造の使用例を図２３及び図２４を用いて説明する。図２３は、マトリックス型データ構造が適用できる自動車保険の例であって、各行には、Ａ、Ｂ、Ｃ、Ｄ及びＥという保険商品名とその保障内容（用途・車種、年

齢別不担保特約、ファミリーバイク特約及び無保険者傷害保険）が示されている。一方各列には、Ａ、Ｂ、Ｃ、Ｄ及びＥのそれぞれについて、用途・車種、年齢別不担保特約、ファミリーバイク特約及び無保険者傷害保険の情報が示されている。

【００６１】

図２３のようなテーブルを表すために、例えば図２４のようなマトリックス型データ構造を規定する。すなわち、保険商品レコード４７２は行部４６２に対応し、保険商品コードがキーであり、商品名という属性が与えられる。保障レコード４７０は列部２６０に対応し、保障コードがキーであり、保障名という属性が与えられる。商品保障組合せレコード４７４はセル部４６４に対応する。保障パターン・レコード４７６は列タイプ部４６６に対応し、保障コード及びパターンＩＤがキーであり、パターン持ち方区分（真偽、文字列、文字列リスト（複数の文字列）、文字列範囲、文字列範囲リスト、数値、数値リスト、数値範囲、数値範囲リストのいずれかを識別するコード）と、パターン持ち方区分で特定された真偽、文字列、文字列リスト（複数の文字列）、文字列範囲、文字列範囲リスト、数値、数値リスト、数値範囲、数値範囲リストのいずれかが属性として与えられる。このように、マトリックス型データ構造のデータ構造部３２０に対応する雛型プログラム３８０には、少なくとも行部４６２、列部４６０、及び列タイプ部４６６に属性を与えるための解決ロジック３８２が含まれている。また、マトリックス型データ構造の場合には、列タイプ部４６６をセル部４６４に割り当てるための解決ロジックも含まれる。

【００６２】

そして、図２３のようなテーブルの場合、保険商品レコード４７２に対応するインスタンスは、Ａ、Ｂ、Ｃ、Ｄ及びＥの５つ存在する。また、保障レコード４７０に対応するインスタンスは、用途・車種、年齢別不担保特約、ファミリーバイク特約、及び無保険者傷害特約の４つ存在する。商品保障組合せレコード４７４に対応するインスタンスは、行及び列の組合せで２０個存在する。保障パターン・レコード４７６に対応するインスタンスは、８つ存在する。これは、用途・車種につき２つ、年齢別不担保特約につき２つ、ファミリーバイク特約につき２

つ、無保険者傷害特約につき２つである。すなわち、自家用普通乗用車・自家用小型乗用車・自家用軽四輪乗用車・二輪車・原付自転車という文字列リストが属性として含まれる用途・車種についてのインスタンスとが含まれる。また、自家用普通乗用車・自家用小型乗用車・自家用軽四輪乗用車という文字列リストが属性として含まれる用途・車種についてのインスタンスと、２１・２６・３０・０という数値リストが属性として含まれる年齢別不担保特約についてのインスタンスと、５０という数値リストが属性として含まれる年齢別不担保特約についてのインスタンスとが含まれる。さらに、Ｙｅｓ（あり）という真偽が属性として含まれるファミリーバイク特約についてのインスタンスと、Ｎｏ（なし）という真偽が属性として含まれるファミリーバイク特約についてのインスタンスとが含まれる。また、Ｙｅｓ（あり）という真偽が属性として含まれる無保険者傷害特約についてのインスタンスと、Ｎｏ（なし）という真偽が属性として含まれる無保険者傷害特約についてのインスタンスとが含まれる。

【００６３】

また、マトリックス型データ構造の操作基本部３１０には、例えば以下に示すような基本操作が含まれる。

（１）列の追加（列タイプ情報なし）

列のインスタンスが追加される。列に対応するセル１列分が追加されるが列タイプへのリレーションは生成されない。列の情報が引数。

（２）列の追加(列タイプ情報付き)

列のインスタンスと列タイプが追加される。列に対応するセル１列分が追加されるが列タイプへのリレーションは生成されない。列の情報が引数。

（３）列の追加（列タイプ情報、セル情報付き）

列のインスタンスと列タイプが追加される。列に対応するセル１列分が追加され列タイプへのリレーションを保持する。列の情報が引数。

（４）列の削除

指定された列を削除する。列を削除するとその列に関連するセル、列タイプも消える。列キーが引数。

（５）列の変更（セル情報なし）

指定された列を変更する（セル情報なし）。列の情報が引数。

（６）列単位セル変更

指定された列をセル情報付きで変更する。列と行のインスタンス数分だけのセル情報が引数。

（７）行の追加（セル情報付き）

行のインスタンスと関連するセル１行分が追加され、列タイプのリレーションを保持する。行の情報が引数。

（８）行の追加（セル情報なし）

行のインスタンスと関連するセル１行分が追加されるが、列タイプのリレーションは生成されない。行の情報が引数。

（９）行の削除

指定された行を削除する。関連するセルも削除される。行キーが引数。

（１０）行の変更（セル情報なし）

指定された行を変更する（セル情報なし）。行の情報が引数。

（１１）セルの変更

指定されたセルの列タイプを変更する。セルの情報が引数。

（１２）行単位セル変更

指定された行をセル情報付きで変更する。行と列のインスタンス数分だけのセル情報が引数。

（１３）列タイプの追加

列タイプのインスタンスを追加する。列タイプの情報が引数。

（１４）列タイプの削除

指定された列タイプを削除する。セルがから参照されている列タイプは削除できない。列タイプの情報が引数。

（１５）列タイプの変更

列タイプの情報を更新する。列タイプの情報が引数。

（１６）セル検索（セル情報なし）

指定された行、列のセルを返す。セルの列タイプ情報が引数。

（１７）行検索（セル情報なし）

引数である行キーに対応する行の情報を返す。

（１８）列検索（セル情報なし）

引数である列キーに対応する列の情報を返す。

（１９）行検索（セル情報付き）

指定された行の情報を返す（セルの中に列タイプのリンクの情報あり）。行キーが引数。

（２０）行検索（条件指定）

引数である条件を指定して、複合の行の集合情報（セルの列タイプ情報含む）を返す。

（２１）列タイプ検索（列のキー指定）

列のキーを指定して列タイプリストを返す。

（２２）行リスト検索（列セル値指定）

指定された列、セル値に該当する行リストを返す。

マトリックス型データ構造に対する基本操作に対応する各雛型プログラムには、対象となるプログラム固有の設定を行うための解決ロジックが埋め込まれている。

【００６４】

５．階層型データ構造の場合

次に階層型データ構造に係るデータ構造解決ユニット３について説明する。階層型データ構造は、組織や分類、集計値などのように、複数のエンティティ（レコードタイプ）が一列の階層構造をもっており、階層の概念が固定的である場合に用いる。最上位以外は、上位のインスタンスが必ず一つ決まる木構造であることが前提となる。インスタンス間で階層に抜けがあってよい。階層の数に制限は無い。

【００６５】

階層型データ構造を選択した場合に生成されるプログラムが出力する画面の一例を図２５に示す。データ構造部３２０には、１段目のデータを表示するトップ部２５１０と、２段目のデータを表示する第２段部２５２０と、３段目のデータを表示する第３段部２５３０と、．．．最下層であるＮ段目のデータを表示する

ボトム部２５４０とが含まれる。また、操作基本部３１０には、１段目にレコードを登録するトップ登録ボタン２５５０（階層のレベル（段数）をトップと指定した登録操作）と、１段目のレコードを更新するトップ更新ボタン２５５２（階層のレベル（段数）をトップと指定した更新操作）、１段目のレコードを削除するトップ削除ボタン２５５４（階層のレベル（段数）をトップと指定した削除操作）、最下層のレコードを登録するボトム登録ボタン２５５６（階層のレベル（段数）をボトムと指定した登録操作）、最下層のレコードを更新するボトム更新ボタン２５５８（階層のレベル（段数）をボトムと指定した更新操作）、最下層のレコードを削除するボトム削除ボタン２５６０（階層のレベル（段数）をボトム指定した削除操作）、あるレコードのルートを検索するルート検索ボタン２５６２、あるレコードの子を検索する子検索ボタン２５６４、あるレコードのリーフを検索するリーフ検索ボタン２５６６等が含まれる。中間の段階のレコードを操作するボタンも図示していないが存在する。

【００６６】

図２５のような画面を出力するようなプログラムが使用する本実施の形態における階層型データ構造は、図２６に示すように、キーと属性とを含む最上位階層のトップ部４９０と、キーと親のキーと属性とを含む第２段部４９２と、キーと親のキーと属性とを含む第３段部４９４と、．．．キーと親のキーと属性とを含む最下層のボトム部４９６と、トップ部４９０から第２段部４９２へのリンクと、第２段部４９２から第３段部４９４へのリンクと、第３段部４９４から第４段部へのリンクと、．．．第Ｎ－１段部からボトム部４９６へのリンクとが含まれる。なお、あるレコードの下位のレコードは複数存在する場合がある。また、トップ部４９０等の各レコード・タイプには、生成、削除、検索等の処理を実行する処理部が設けられる。

【００６７】

さらに、操作基本部３１０には、以下に示すような基本操作が含まれる。なお括弧内の文字列はメソッド名である。

（１）ノード検索（findNodeByKey）

ノードのキーを指定して、ノードを検索する。

（２）上位ノード検索（findUpperOfNode）

ノードのキーを指定して、そのノードの上位ノードを検索する。いくつ上のノードかは、相対的な位置関係で指定する。

（３）ルートノード検索（findRootOfNode）

ノードのキーを指定して、そのノードのルートノードを検索する。

（４）下位ノード検索

ノードのキーを指定して、そのノードの下位ノードを検索する。いくつ下のノードまで検索するかは、相対的な位置関係で指定する（findLowersOfNode）。そして、検索されたノードリストから、個数を指定して、ノード情報を取り出して、返却値を作る（nextNodes）。

（５）リーフノード検索

ノードのキーを指定して、そのノードのリーフノードを検索する（findLeavesOfNode）。そして、検索されたノードリストから、個数を指定して、ノード情報を取り出して、返却値を作る（nextNodes）。

（６）ルートに至るノード検索

ノードのキーを指定して、そのノードからルートに至るまでのノードを検索する（findUppersOfNode）。そして、検索されたノードリストから、個数を指定して、ノード情報を取り出して、返却値を作る（nextNodes）。

（７）絶対階層ノード検索

ノードのキーを指定して、そのノードにリンクされている、指定した絶対階層に位置するノードを検索する（findNLevelNodesOfNode）。そして、検索されたノードリストから、個数を指定して、ノード情報を取り出して、返却値を作る（nextNodes）。

（８）絶対階層ルート検索

絶対階層を指定して、その階層の中でルートになっているノードを検索する（findRootsByLevel）。そして、検索されたノードリストから、個数を指定して、ノード情報を取り出して、返却値を作る（nextNodes）。

（９）絶対階層リーフ検索

絶対階層を指定して、その階層の中でリーフになっているノードを検索する（

findLeavesByLevel）。そして、検索されたノードリストから、個数を指定して、ノード情報を取り出して、返却値を作る（nextNodes）。

（１０）絶対階層全ノード検索

絶対階層を指定して、その階層の中のノードを全て検索する（findNodesByLevel）。そして、検索されたノードリストから、個数を指定して、ノード情報を取り出して、返却値を作る（nextNodes）。

（１１）ノード条件検索

ノードを条件指定で検索する。そして、検索されたノードリストから、個数を指定して、ノード情報を取り出して、返却値を作る（nextNodes）。

（１２）キー付きでノード生成（createNodeWithKey1）

キーを指定して、ノードを新規に生成する。上位ノードが指定されていれば、その下位ノードになる。返却値は真偽。

（１３）キー付きでノード生成

キーを指定して、ノードを新規に生成する。上位ノードが指定されていれば、その下位ノードになる（createNodeWithKey2）。そして、生成したノードリストから、個数を指定して、ノード情報を取り出して、返却値を作る（nextNodes）。

（１４）キー無しでノード生成

キーを指定しないで、ノードを新規に生成する（createNodeWithoutKey）。キーは採番サービスから取得する。上位ノードが指定されていれば、その下位ノードになる。そして、生成したノードリストから、個数を指定して、ノード情報を取り出して、返却値を作る（nextNodes）。

（１５）ノードの削除（removeNode）

キーで指定したノードを削除する。リンクしていた上位ノードと下位ノードは残り、リンクのみ消える。

（１６）ノードを削除してリンク（removeAndLink）

キーで指定したノードを削除し、その上位ノードと下位ノードをリンクする。

（１７）ノードを削除してシフト（removeAndShift）

キーで指定したノードを削除し、その配下のノードを、削除したノードと同じ

階層の別のノードへ移動する。

（１８）ノードを階層ごと削除（removeHierarchy）

キーで指定したノードを、その配下のノードごと削除する。

（１９）ノードをリンクする（linkNodes）

キーで指定した二つのノードをリンクする。

（２０）ノード間のリンクを切る（unlinkNodes）

キーで指定した二つのノードのリンクを切る。

（２１）ノード情報の更新

ノードの情報を更新する。返却値は真偽の場合とノードの情報の場合がある。

【００６８】

階層型データ構造に対してこのようなデータ構造解決ユニット３が用意されており、解決ロジック解析部７はデータ構造解決ユニット３の解決ロジックを解析して、例えば図２７に示すような解決情報入力画面１１ｊを作成してユーザに出力する。

【００６９】

解決情報入力画面１１ｊには、データ構造解決ユニット３のデータ構造部３２０に対する解決情報を入力するためのデータ構造解決部２７００乃至２７２０と、データ構造解決ユニット３の操作基本部３１０に対応する雛型プログラム３８０に埋め込まれた解決ロジック３８２に対する解決情報を入力するための操作解決部２７３０とが含まれる。

【００７０】

データ構造解決部２７００は、最上位階層であるトップ部４９０の解決情報を入力する部分である。ここには、項目名、キーか否か、データの型を入力するようになっている。すなわち、雛型プログラム３８０にはトップ部４９０に属性を与えるための解決ロジック３８２が含まれている。図２７の例では、項目名「事業部コード」が入力されており、この事業部コードはキーであり且つ整数型のデータであることが入力されている。項目名「事業部名称」が入力されており、この事業部名称は文字列型のデータであることが入力されている。項目名「事業部長」が入力されており、この事業部長は文字列型のデータであることが入力され

ている。

【００７１】

データ構造解決部２７１０は、中段部（第２段部乃至第Ｎ－１段部）の解決情報を入力する部分である。ここには、項目名、キーか否か、データの型を入力するようになっている。すなわち、雛型プログラム３８０には中段部に属性を与えるための解決ロジック３８２が含まれている。図２７の例では、項目名「部コード」が入力されており、この部コードはキーであり且つ整数型のデータであることが入力されている。項目名「部名称」が入力されており、この部名称は文字列型のデータであることが入力されている。項目名「事業部コード」が入力されており、この事業部コードは整数型のデータであることが入力されている。

【００７２】

データ構造解決部２７２０は、最下層であるボトム部４９６の解決情報を入力する部分である。ここには、項目名、キーか否か、データの型を入力するようになっている。すなわち、雛型プログラム３８０にはボトム部４９６に属性を与えるための解決ロジック３８２が含まれている。図２７の例では、項目名「課コード」が入力されており、この課コードはキーであり且つ整数型のデータであることが入力されている。項目名「課名称」が入力されており、この課名称は文字列型のデータであることが入力されている。項目名「部コード」が入力されており、この事業部長は文字列型のデータであることが入力されている。

【００７３】

操作解決部２７３０は、データ構造解決ユニット３の操作基本部３１０に対応する雛型プログラム３５０乃至３７０に埋め込まれた解決ロジック３５２に対する解決情報を入力する部分である。まず、操作名を最初に入力するようにユーザに求める。図２７では操作名は［部署追加］と入力されている。この部署追加に対応する基本操作を次に選択する。この選択には基本操作名リストが用いられる。図２７では部署追加に対応する基本操作は［ノード追加］と選択されている。この基本操作「ノード追加」に対応する雛型プログラムが読み出され、埋め込まれた解決ロジックに代わって入力されるべき解決情報の入力が求められる。図２７では、対象ノード名という項目に対して［部］が入力されている。追加親情報

という項目に対して［ネットワーク事業部］が入力されている。エラーメッセージという項目に対して［部コードが重複しています］が入力されている。固有チェックという項目に対して［地区コードが一致しているか？］が入力されている。固有チェックは、チェックの条件を論理式・算術式で記述可能である。

【００７４】

また、次の操作名は［課検索］であり、これに対応する基本操作が［ノード検索］であることが示されている。このような入力処理が、対象となるプログラムの仕様に沿って必要な操作を規定し終えるまで繰り返される。

【００７５】

６．ツリー型データ構造の場合

ツリー型データ構造は、分類やディレクトリのように、上位下位の関係を一つのエンティティで管理している場合に用いられる。各インスタンス間は上位が一つ決まる木構造だが、階層の数や概念が固定ではなく、予め決められない場合に用いる。

【００７６】

ツリー型データ構造を選択した場合に生成されるプログラムが出力する画面の一例を図２８に示す。データ構造部３２０には、ノードエンティティ情報２８００が含まれる。例えば、ディレクトリ構造の図が示される。また、操作基本部３１０には、ノードを登録するためのノード登録ボタン２８１０、ノードを更新するためのノード更新ボタン２８２０、ノードを削除するためのノード削除ボタン２８３０、親ノードを変更するための親コード変更ボタン２８４０、ツリー展開のためのボタン２８５０、親ノードを検索するための親ノード検索ボタン２８６０、ルーフ検索のためのボタン２８６０等が含まれる。

【００７７】

本実施の形態における階層型データ構造は、図２９に示すように、ノード・レコード２９００だけで構成される。このノード・レコード２９００には、キーと、その上位ノードのキー（最上位の場合には０、それ以下のノードはＮ個（Ｎは０以上の整数）と、属性が含まれる。なお、ノード・レコードには、生成、削除、検索等を実行する処理部が設定される。

【００７８】

また、ツリー型データ構造の操作基本部３１０には、以下に示すような基本操作が含まれる。なお括弧内の文字列はメソッド名である。

（１）ノード検索（findNodeByKey）

ノードのキーを指定して、ノードを検索する。

（２）親ノード検索（findParentOfNode）

ノードのキーを指定して、そのノードの親ノードを検索する。

（３）ルートノード検索（findRootOfNode）

ノードのキーを指定して、そのノードのルートノードを検索する。

（４）子ノード検索

ノードのキーを指定して、そのノードの子ノードを検索する（findChildrenOfNode）。そして、検索されたノードリストから、１つのノード情報を取り出して、返却値を作る（nextNode）。又は、検索されたノードリストから、個数を指定して、ノード情報を取り出して、返却値を作る（nextNodes）。

（５）リーフノード検索

ノードのキーを指定して、そのノードのリーフノードを検索する（findLeavesOfNode）。検索されたノードリストから、１つのノード情報を取り出して、返却値を作る（nextNode）。又は、検索されたノードリストから、個数を指定して、ノード情報を取り出して、返却値を作る（nextNodes）。

（６）ルートに至るノード検索

ノードのキーを指定して、そのノードからルートへ向けて上位ノードを検索する（findAncestorsOfNode）。いくつ上のノードまで検索するかは、段数で指定する。そして、検索されたノードリストから、１つのノード情報を取り出して、返却値を作る（nextNode）。又は、検索されたノードリストから、個数を指定して、ノード情報を取り出して、返却値を作る（nextNodes）。

（７）リーフに至るノード検索

ノードのキーを指定して、そのノードからリーフへ向けて下位ノードを検索する（findTreeOfNode）。いくつ下のノードまで検索するかは、段数で指定する。そして、検索されたノードリストから、１つのノード情報を取り出して、返却値

を作る（nextNode）。又は、検索されたノードリストから、個数を指定して、ノード情報を取り出して、返却値を作る（nextNodes）。

（８）ノード条件検索

ノードを条件指定で検索する。そして、検索されたノードリストから、１つのノード情報を取り出して、返却値を作る（nextNode）。検索されたノードリストから、個数を指定して、ノード情報を取り出して、返却値を作る（nextNodes）。

（９）子ノード個数検索（countChildren）

ノードのキーと段数を指定して、その段数までの子ノードの個数を取得する。

（１０）キー付きでノード生成（createWithKey1(createNode)）

キーを指定して、ルート・リーフノードを新規に生成する。返却値は真偽。

（１１）キー付きでノード生成（createWithKey2(createNode)）

キーを指定して、ルート・リーフノードを新規に生成する。返却値は生成したノードの情報。

（１２）キー無しでノード生成（createWithoutKey(createNode)）

キーを指定しないで、ルート・リーフノードを新規に生成する。返却値は生成したノードの情報。キーは採番サービスから取得する。

（１３）キー付きでルートノード生成（createRootWithKey1(createRoot)）

キーを指定して、ルートノードを新規に生成する。返却値は真偽。

（１４）キー付きでルートノード生成（createRootWithKey2(createRoot)）

キーを指定して、ルートノードを新規に生成する。返却値は生成したノードの情報。

（１５）キー無しでルートノード生成（createRootWithoutKey(createRoot)）

キーを指定しないで、ルートノードを新規に生成する。返却値は生成したノードの情報。キーは採番サービスから取得する。

（１６）キー付きでリーフノード生成（createLeafWithKey1(createLeaf)）

キーを指定して、リーフノードを新規に生成する。返却値は真偽。

（１７）キー付きでリーフノード生成（createLeafWithKey2(createLeaf)）

キーを指定して、リーフノードを新規に生成する。返却値は生成したノードの

情報。

（１８）キー無しでリーフノード生成（createLeafWithoutKey(createLeaf)）

キーを指定しないで、リーフノードを新規に生成する。返却値は生成したノードの情報。キーは採番サービスから取得する。

（１９）キー付きでツリー生成（createTreeWithKey1）

キーを指定して、ツリーを新規に生成する。返却値は真偽。親ノードが指定されていれば、そのノードのサブツリーになる。

（２０）キー付きでツリー生成

キーを指定して、ツリーを新規に生成する（createTreeWithKey2）。親ノードが指定されていれば、そのノードのサブツリーになる。そして、生成されたノードリストから、１つのノード情報を取り出して、返却値を作る（nextNode）。又は、生成したノードリストから、個数を指定して、ノード情報を取り出して、返却値を作る（nextNodes）。

（２１）キー無しでツリー生成

キーを指定しないで、ツリーを新規に生成する（createTreeWithoutKey）。キーは採番サービスから取得する。親ノードが指定されていれば、そのノードのサブツリーとなる。そして、検索されたノードリストから、１つのノード情報を取り出して、返却値を作る（nextNode）。又は、生成したノードリストから、個数を指定して、ノード情報を取り出して、返却値を作る（nextNodes）。

（２２）ノードの削除（removeNode）

キーで指定したノードを削除する。リンクしていた親ノードと子ノードは残り、リンクのみ消える。

（２３）ノードを削除してリンク（removeAndLink）

キーで指定したノードを削除し、その親ノードと子ノードをリンクする。

（２４）ノードからリーフまで削除（removeTree）

キーで指定したノードから、そのノードのリーフノードまで削除する。

（２５）ノードをリンクする（linkNodes）

キーで指定した二つのノードをリンクする。

（２６）ノード間のリンクを切る（unlinkNodes）

キーで指定した二つのノードのリンクを切る。

（２７）親ノードを変更する（changeLink）

キーで指定したノードとその親ノードとのリンクを変更する。

（２８）ノード情報の更新

ノードの情報を更新する。返却値は真偽の場合と、返却値はノードの情報の場合がある。

【００７９】

ツリー型データ構造に対してこのようなデータ構造解決ユニット３が用意されており、解決ロジック解析部７はデータ構造解決ユニット３の解決ロジックを解析して、例えば図３０に示すような解決情報入力画面１１Ｋを作成してユーザに出力する。

【００８０】

解決情報入力画面１１Ｋには、データ構造解決ユニット３のデータ構造部３２０に対する解決情報を入力するためのデータ構造解決部３０００と、データ構造解決ユニット３の操作基本部３１０に対応する雛型プログラム３８０に埋め込まれた解決ロジック３８２に対する解決情報を入力するための操作解決部３０１０とが含まれる。

【００８１】

データ構造解決部３０００は、ノード・レコード２９００の解決情報を入力する部分である。ここには、項目名、キーか否か、データの型を入力するようになっている。すなわち、雛型プログラム３８０にはノード・レコード２９００に属性を与えるための解決ロジック３８２が含まれている。図３０の例では、項目名「職制コード」が入力されており、この職制コードはキーであり且つ整数型のデータであることが入力されている。項目名「職制名称」が入力されており、この職制名称は文字列型のデータであることが入力されている。項目名「上位職制コード」が入力されており、この上位職制コードは整数型のデータであることが入力されている。

【００８２】

操作解決部３０１０は、データ構造解決ユニット３の操作基本部３１０に対応

する雛型プログラム３５０乃至３７０に埋め込まれた解決ロジック３５２に対する解決情報を入力する部分である。まず、操作名を最初に入力するようにユーザに求める。図３０では操作名は［職制変更］と入力されている。この職制変更に対応する基本操作を次に選択する。この選択には基本操作名リストが用いられる。図３０では職制変更に対応する基本操作は［ノード変更］と選択されている。この基本操作「ノード変更」に対応する雛型プログラムが読み出され、埋め込まれた解決ロジックに代わって入力されるべき解決情報の入力が求められる。図３０では、対象ノード名という項目に対して［部］が入力されている。追加親情報という項目に対して［ネットワーク事業部］が入力されている。エラーメッセージという項目に対して［職制コードが重複しています］が入力されている。固有チェックという項目に対して［部員が存在しているか？］が入力されている。固有チェックは、チェックの条件を論理式・算術式で記述可能である。

【００８３】

また、次の操作名は［職制検索］であり、これに対応する基本操作が［ノード検索］であることが示されている。このような入力処理が、対象となるプログラムの仕様に沿って必要な操作を規定し終えるまで繰り返される。

【００８４】

７．明細主導伝票型データ構造の場合

明細主導伝票型データ構造は、伝票型業務だが、ヘッダ単位ではなく、明細単位の情報が中心になる場合に用いられる。業務の推移は、新しく伝票を起こすのではなく、明細を束ねる単位を変えて新しいヘッダを付けていくことで行う。明細の実体はそのままで業務が流れていく。状態変化を握るのはヘッダではなく明細で、状態の変化の結果ヘッダが変わることがある。

【００８５】

例えば、図３１（ｄ）のような明細番号１００１乃至１００４という取引明細が存在する場合に、これに対して売掛番号１０１の売掛伝票（図３１（ａ））が生成される。そして、明細番号１００１乃至１００３についてＡ商店にて検収されると、請求番号２０１の請求伝票（図３１（ｂ））が生成される。そして、明細番号１００１乃至１００３についての請求伝票に対する振込みが確認されると

、入金番号３０１の入金伝票（図３１（ｃ））が生成される。明細番号１００１乃至１００３までが請求処理に回されても、売掛伝票の見え方は変化せず、売掛伝票の明細は、明細番号１００１乃至１００４までである。このように、取引明細は変化せず、新たなヘッダとなる売掛伝票、請求伝票、入金伝票が取引明細に関連して付加されていく。このような業務に対して明細主導伝票型データ構造が有用である。

【００８６】

次に、明細伝票型データ構造を選択した場合に生成されるプログラムが出力する画面の一例を図３２に示す。データ構造部３２０には、ヘッダ部１（例えば売掛伝票）のデータ表示部３２００と、ヘッダ部２（例えば請求伝票）のデータ表示部３２１０と、．．．．ヘッダ部Ｎ（例えば入金伝票）のデータ表示部３２２０と、明細部（例えば取引明細）のデータ表示部３２３０とが含まれる。また、操作基本部３１０には、ヘッダ１登録のためのボタン３２４０、ヘッダ１明細追加のためのボタン３２５０、ヘッダ１をヘッダ２へ切り換えるためのボタン３２６０、明細更新のためのボタン３２７０等が含まれる。

【００８７】

図３２のような画面を出力するようなプログラムが使用する本実施の形態における階層型データ構造は、図３３に示すように、ヘッダ部１（３３００）と、ヘッダ部２（３３１０）と、ヘッダ部３（３３２０）と、．．．明細部３３３０と、ヘッダ部１（３３００）と明細部３３３０とのリンクと、ヘッダ部２（３３１０）と明細部３３３０とのリンクと、ヘッダ部３（３３２０）と明細部３３３０とのリンクとが含まれる。なお、ヘッダ部１（３３００）には、キー１と属性とが含まれる。ヘッダ部２（３３１０）には、キー２と属性とが含まれる。ヘッダ部３（３３２０）には、キー３と属性とが含まれる。明細部３３３０には、明細部３３３０のキーと、ヘッダ部１（３３００）のキー１と、ヘッダ部２（３３１０）のキー２と、ヘッダ部３（３３２０）のキー３と、属性とが含まれる。なお、ここでは３つヘッダが存在する例を示したが、ヘッダの数は任意である。

【００８８】

また、ヘッダ部１（３３００）、ヘッダ部２（３３１０）、及びヘッダ部３（

３３２０）の各々には、生成、削除、キー検索及び全件検索を実行する処理部が設けられる。明細部３３３０には、生成、削除及びキー検索等を実行する処理部が設けられる。

【００８９】

さらに、明細主導伝票型データ構造の操作基本部３１０には、以下に示すような基本操作が含まれる。なお括弧内の文字列はメソッド名である。

（１）伝票検索（findSlipByHeaderKey）

指定したヘッダのキーを持つ伝票を検索する。

（２）明細１件検索（findDetailByKey）

指定したヘッダのキーを持つ伝票の明細を行番号を指定して１件検索する。

（３）明細状態検索

指定した状態をもつ明細を検索する（findDetailByState）。そして、検索後に１件分の明細情報を取得する（NextDetail）。

（４）ヘッダ種指定ヘッダ検索（findOldHeaderByKind）

明細と伝票ヘッダ種を指定してヘッダを検索する。

（５）ヘッダ種指定伝票検索（findOldSlipByKind）

明細と伝票ヘッダ種を指定して伝票を検索する。

（６）条件による伝票検索

伝票を条件指定で検索する。そして、条件検索後に伝票１件分の全情報を取得する（nextSlip）。

（７）キーつき伝票新規作成（createSlipWithKey1）

キーを設定して伝票を新規作成し、結果を真偽で返す。

（８）キーつき伝票新規作成

キーを設定して伝票を新規作成する（createSlipWithKey2）。そして、生成した伝票の情報を返す。

（９）キーなし伝票新規作成

キーを設定せずに伝票を新規作成する（createSlipWithoutKey）。そして、生成した伝票の情報を返す。

（１０）明細追加（insertDetails1）

既存の伝票に明細を複数件追加し、結果を真偽で返す。

（１１）明細追加

既存の伝票に明細を複数件追加する（insertDetails2）。追加した伝票の情報を返す。

（１２）伝票ヘッダ種変更（changeHeaderWithKey）

明細の状態を指定された状態に更新し、キーと伝票ヘッダ種で指定されたヘッダに変更する。指定された伝票種のヘッダが存在しない場合には、ヘッダを新規に生成する。

（１３）伝票ヘッダ種変更（changeHeaderWithoutKey）

明細の状態を指定された状態に更新し、指定された伝票ヘッダ種のヘッダを新規に作成し、ヘッダを変更する。

（１４）明細状態変更（changeState）

明細の状態を指定された状態に変更する。明細状態クラスに設定された遷移可能先、削除許可フラグのチェックを行なう。

（１５）明細削除（removeDetail）

明細を１件削除する。

（１６）伝票削除（removeSlip）

伝票を削除する。

（１７）ヘッダ更新

キーで指定したヘッダの内容を更新し、結果を真偽で返す。

（１８）ヘッダ更新

キーで指定したヘッダの内容を更新し、更新したヘッダの内容を返す。

（１９）明細更新

キーで指定した伝票の明細の内容を更新し、結果を真偽で返す。

（２０）明細更新

キーで指定した伝票の明細の内容を更新し、更新した伝票の内容を返す。

【００９０】

明細主導伝票型データ構造に対してこのようなデータ構造解決ユニット３が用意されており、解決ロジック解析部７はデータ構造解決ユニット３の解決ロジッ

クを解析して、例えば図３４に示すような解決情報入力画面１１１を作成してユーザに出力する。

【００９１】

解決情報入力画面１１１には、データ構造解決ユニット３のデータ構造部３２０に対する解決情報を入力するためのデータ構造解決部３４００乃至３４２０と、データ構造解決ユニット３の操作基本部３１０に対応する雛型プログラム３８０に埋め込まれた解決ロジック３８２に対する解決情報を入力するための操作解決部３４３０とが含まれる。

【００９２】

データ構造解決部３４００は、ヘッダ部１（３３００）の解決情報を入力する部分である。ここには、項目名、キーか否か、データの型を入力するようになっている。すなわち、雛型プログラム３８０にはヘッダ部１（３３００）に属性を与えるための解決ロジック３８２が含まれている。図３４の例では、項目名「売掛番号」が入力されており、この売掛番号はキーであり且つ整数型のデータであることが入力されている。項目名「受注日」が入力されており、この受注日は文字列型のデータであることが入力されている。項目名「顧客名」が入力されており、この顧客名は文字列型のデータであることが入力されている。また、項目名「合計額」が入力されており、この合計額は整数型のデータであることが入力されている。

【００９３】

データ構造解決部３４１０は、ヘッダ部２（３３１０）の解決情報を入力する部分である。ここには、項目名、キーか否か、データの型を入力するようになっている。すなわち、雛型プログラム３８０にはヘッダ部２（３３１０）に属性を与えるための解決ロジック３８２が含まれている。図３４の例では、項目名「請求番号」が入力されており、この請求番号はキーであり且つ整数型のデータであることが入力されている。項目名「請求日付」が入力されており、この請求日付は文字列型のデータであることが入力されている。項目名「請求先」が入力されており、この請求先は文字列型のデータであることが入力されている。項目名「合計額」が入力されており、この合計額は整数型のデータであることが入力され

ている。

【００９４】

データ構造解決部３４２０は、明細部３３３０の解決情報を入力する部分である。ここには、項目名、キーか否か、データの型を入力するようになっている。すなわち、雛型プログラム３８０には明細部３３３０に属性を与えるための解決ロジック３８２が含まれている。図３４の例では、項目名「取引番号」が入力されており、この取引番号はキーであり且つ整数型のデータであることが入力されている。項目名「商品」が入力されており、この商品は文字列型のデータであることが入力されている。項目名「数量」が入力されており、この数量は整数型のデータであることが入力されている。項目名「単価」が入力されており、この単価は整数型のデータであることが入力されている。

【００９５】

操作解決部３４３０は、データ構造解決ユニット３の操作基本部３１０に対応する雛型プログラム３５０乃至３７０に埋め込まれた解決ロジック３５２に対する解決情報を入力する部分である。まず、操作名を最初に入力するようにユーザに求める。図３４では操作名は［売掛伝票登録］と入力されている。この売掛伝票登録に対応する基本操作を次に選択する。この選択には基本操作名リストが用いられる。図３４では売掛伝票登録に対応する基本操作は［伝票登録］と選択されている。この基本操作「伝票登録」に対応する雛型プログラムが読み出され、埋め込まれた解決ロジックに代わって入力されるべき解決情報の入力が求められる。図３４では、対象ヘッダ名という項目に対して［売掛］が入力されている。追加親情報という項目に対して［顧客名］が入力されている。エラーメッセージという項目に対して［取扱商品がありません］が入力されている。固有チェックという項目に対して［その顧客の限度額を超えていないか？］が入力されている。固有チェックは、チェックの条件を論理式・算術式で記述可能である。このように、基本操作に対応する本雛型プログラムには、ヘッダ・レコードの状態を操作との関連で定義するための解決ロジックと、プログラムの仕様に従った設定をレコードの属性、レコードの状態、若しくはレコードの属性及び状態の組合せで記述するための解決ロジックが埋め込まれている。

【００９６】

また、次の操作名は［請求発行］であり、これに対応する基本操作が［ヘッダ切替］であることが示されている。このような入力処理が、対象となるプログラムの仕様に沿って必要な操作を規定し終えるまで繰り返される。

【００９７】

８．構成型データ構造の場合

構成型データ構造は、部品表のように、構成するものと構成されるものを一つのエンティティで取り扱い、そのｎ対ｍの関連（構成情報）を管理するような場合に用いられる。１対ｎの場合にはツリー型データ構造を用いる。

【００９８】

例えば図３５に示すように、ＬＸ２７７ＡＡＡという型番について、同一型番の中で版（バージョン）が存在する。ここでは、ＬＸ２７７ＡＡＡ－０１０（日付：１２／１０、互換フラグ：ＯＫ、代替グループＩＤ：－－－）という版と、ＬＸ２７７ＡＡＡ－０２０（日付：１２／１１、互換フラグ：ＯＫ、代替グループＩＤ：－－－）という版とが存在している。この３つの版は同一型番ということができる。さらに、各版に対して部品が定義できる。図３５ではＬＸ２７７ＡＡＡ－０１０という版に対して、ＸＸＸ－０１０（１２／１０版）が５つと、ＹＹＹ－０１０（１０／１０版）と、ＺＺＺ－０３０（１２／１０版）とが含まれる。この版と部品の間を結ぶリンクが「構成」と呼ばれる。なお、部品についても版が存在する場合もある。その際、その版が基本となる版（基本型番）に対して互換性があるか否かが互換性フラグで表される。「ＯＫ」は互換性があることを示している。代替グループＩＤは、代替部品が存在する場合には、その代替部品のグループのＩＤである。

【００９９】

図３５のような状態を表すための構成型データ構造は、図３６に示すように、型番（ノード）部３６００と、型番版数（ノード版数）部３６１０と、構成部３６２０と、型番部３６００と型番版数部３６１０とのリンクと、型番版数３６１０と構成部３６２０とのリンクとが含まれる。型番部３６００には、キーである型番と、属性である型番情報とが含まれる。型番版数部３６１０には、キーであ

る型番及び日付と、属性である代替グループＩＤ、互換部品フラグ及び部品情報とが含まれる。また、構成部３６２０には、キーである上位ノード及び下位ノードと、属性である構成情報、上限個数及び下限個数とが含まれる。型番部３６００に対して型番版数部３６１０はＮ個（Ｎは０以上の整数）存在する。１つの型番版数３６１０について、基本型番である型番部３６００は存在することもあるし、存在しない場合もある。型番版数３６１０に対して構成部３６２０は、Ｍ個（Ｍは０以上の整数）存在する。但し、上位の型番版数３６１０は存在する場合もあれば存在しない場合もある。同様に、下位の型番版数３６１０は存在する場合もあれば存在しない場合もある。なお、構成型データ構造のデータ構造部３２０に対応する雛型プログラム３８０には、型番部３６００と、型番版数部３６１０と、構成部３６２０とに属性を与えるための解決ロジック３８２が含まれる。

【０１００】

また、構成型データ構造の操作基本部３１０には、以下に示すような基本操作が含まれる。なお括弧内の文字列はメソッド名である。

（１）ノード版数検索（findNodeRevisionByKey）

プライマリキーによるノード版数１件検索。

（２）ノード版数複数件検索

指定されたノード版数と同じノードを参照するノード版数を検索する（findNodeRevisionsByNode）。そして、検索結果リストからメッセージリストを取り出す（nextNodeRevision）。

（３）交換可能ノード版数複数件検索

指定したノードと交換可能なノード版数を検索する（findReplaceableNodeRevisionsByNode）。そして、検索結果リストからメッセージリストを取り出す（nextNodeRevision）。

（４）交換可能チェック（isNodeRevisionReplaceable）

ノード版数が交換可能かをチェックする。

（５）基本版数検索（findBaseNodeRevisionByNode）

ノードのキーにより基本版数を検索する。

（６）基本ノード版数検索（findBaseNodeRevisionByNodeRevision）

ノード版数により基本ノード版数を検索する。

（７）代替可能ノード版数の複数件検索

検索結果リストからメッセージリストを取り出す（findSubstituteRevisionsByNodeRevision）。そして、検索結果リストからメッセージリストを取り出す（nextNodeRevision）。

（８）代替可能確認

指定ノード版数が、他のノード版数と代替可能かどうかを確認する。

（９）最新ノード版数の一件検索（findLatestRevisionByNodeKey）

最新ノード版数の一件検索。

（１０）構成情報のキーによる一件検索（findCompositionByKey）

構成情報のキーによる一件検索。

（１１）上位ノード版数による構成検索

上位ノード版数による構成検索（findCompositionsByUpprNodeRevision）。そして、検索結果リストからメッセージリストを取り出す（nextComposition）。

（１２）指定ノード版数情報による下位ノード版数複数件検索

指定ノード版数情報による下位ノード版数複数件検索（findLowersOfNodeRevision）。そして、検索結果リストからメッセージリストを取り出す（nextNodeRevision）。

（１３）構成情報の下位ノードキーによる複数件検索

構成情報の下位ノードキーにより検索する（FindCompositionsByLowerNodeRevision）。そして、検索結果リストからメッセージリストを取り出す（nextComposition）。

（１４）指定ノード版数情報による上位ノード複数件検索

指定されたノード版数情報により上位ノードを検索する（FindUppersOfNodeRevision）。そして、検索結果リストからメッセージリストを取り出す（nextNodeRevision）。

（１５）ルートノード版数検索

指定ノード版数を基点に全ルートノード版数を検索（findRootsOfNodeRevision）。そして、検索結果リストからメッセージリストを取り出す（nextNodeRevis

ion）。

（１６）リーフノード版数検索

指定ノード版数を基点に全リーフノード版数を検索する（findLeavesOfNodeRevision）。そして、検索結果リストからメッセージリストを取り出す（nextNodeRevision）。

（１７）上方ノード版数展開（expandNodeRevisionUpper）

指定ノード版数を起点に展開レベル階層分の上方ノード版数展開を行う。

（１８）下方ノード版数展開（expandNodeRevisionLower）

指定ノード版数を起点に展開レベル階層分の下方ノード版数展開を行う。

（１９）上方構成情報展開（ExpandCompositionUpper）

指定ノード版数を起点に展開レベル階層分の上方構成情報展開を行う。

（２０）下方構成情報展開（ExpandCompositionLower）

指定ノード版数を起点に展開レベル階層分の下方構成情報展開を行う。

（２１）指定ノード版数が指定構成情報の上位に存在するかをチェック（isNodeRevisionInCompositionUpper）

指定ノード版数が指定構成情報の上位に存在するかをチェックする。

（２２）あるノード版数が、指定したノード版数の上位に存在するかチェック（isNodeRevisionInRevisionUpper）

あるノード版数が、指定したノード版数の上位に存在するかチェックする。

（２３）ノード版数が指定構成情報の下位に存在するかをチェック（isNodeRevisionInCompositionLower）

指定ノード版数が指定構成情報の下位に存在するかをチェックする。

（２４）あるノード版数が、指定したノード版数の下位に存在するかチェック（isNodeRevisionInRevisionLower）

あるノード版数が、指定したノード版数の下位に存在するかチェックする。

（２５）ノード版数登録（キーあり）（createNodeRevisionWithKey1）

ノード版数の登録キーあり。エラーの有無を通知する。キーを指定してノード版数を新規登録する。

（２６）ノード版数登録（キーあり）（登録ノード版数を表示）（createNodeRe

visionWithKey2)

ノード版数の登録 キーあり。登録したノード版数の内容を表示する。キーを指定してノード版数を新規登録する。

（２７）ノード版数登録（キーなし）（createNodeRevisionWithoutKey)

ノード版数の登録キーなし。採番機能で得たキーを用いて、ノード版数を新規登録する。

（２８）構成の登録（返却値は真偽）（createComposition1)

上位ノード版数、下位ノード版数を指定して構成を登録する。

（２９）構成の登録（返却値はノード版数情報）（createComposition2)

上位ノード版数、下位ノード版数を指定して構成を登録する。

（３０）２つのノード版数に対する構成情報の登録（linkNodeRevision)

指定した２つのノード版数に対する関連情報を作成する。

（３１）基本ノード版数及び交換可能版数設定（establishBaseNodeRevision)

基本ノード版数及び交換可能版数の設定。

（３２）基本ノード版数及び交換可能版数の解除（releaseBaseNodeRevision)

基本ノード版数及び交換可能版数の解除。

（３３）交換可能設定（establishReplaceableNodeRevision)

指定ノード版数を交換可能に設定する。

（３４）交換可能状態解除（releaseReplaceableNodeRevision)

指定ノード版数の交換可能状態を解除する。

（３５）ノード版数の更新（返却値は真偽）

ノード版数の更新。返却値はノード版数情報の場合と、真偽の場合とがある。

（３６）構成情報の下位ノード版数変更

指定構成情報の下位側ノード版数を指定されたノード版数に変更する。

（３７）ノード版数一件削除（removeNodeRevision)

ノード版数一件削除。

（３８）構成情報一件削除（removeComposition)

構成情報一件削除。

このような構成型データ構造に対する基本操作に対応する各雛型プログラムに

は、対象となるプログラム固有の設定を行うための解決ロジックが埋め込まれている。

【０１０１】

９．座席予約型データ構造

座席予約型データ構造は、列車や飛行機の予約や、コンサートのチケットなど、資源とその利用機会に対する予約を管理するために用いられる。図３７に飛行機の例を示す。図３７の各行は、飛行機の各機種座席を表すものであり、「７４７－１Ａ」「７４７－１Ｂ」．．．は機種名及び座席記号である。座席は資源ということもできる。一方、各列は便を表したものであり、「１０／１　Ｃ１４」「１０／１　Ｃ１５」．．．は日にち及び便名である。便は機会ということもできる。各座席と各便の組合せは、オカーレンスと呼ばれるが、実際は存在しないものもある。図３７で×印は、実際は存在しない席を表す。また、黒バーは予約の入った座席を表し、黒バーを囲む矩形は予約を表す。空席は、点線で表される。通常、予約については、往復予約等のため便をまたぐ座席予約も可能である。

【０１０２】

このような業務を取り扱うために、図３８のような座席予約型データ構造を用いる。座席予約型データ構造は、機会を表す便部３８００と、資源のグループを表す機種部３８１０と、資源を表す機種座席部３８２０と、オカーレンスを表す座席部３８３０と、予約を表す予約部３８４０と、便部３８００と機種部３８１０とのリンクと、機種部３８１０と座席部３８２０とのリンクと、便部３８００と座席部３８３０とのリンクと、座席部３８２０と座席部３８３０とのリンクと、座席部３８３０と予約部３８４０とのリンクとが含まれる。機種部３８１０１つに対して便部３８００はＮ個（Ｎは０以上の整数）存在する。また、機種部３８１０１つに対して機種座席部３８２０はＭ個（Ｍは０以上の整数）存在する。便部３８００１つに対して、座席部３８３０はＫ個（Ｋは０以上の整数）存在し、予約部３８４０１つに対して座席部３８３０はＬ個（Ｌは０以上の整数）存在する。

【０１０３】

便部３８００には、キーである出発日及び便名と、属性である出発地、到着地

、予定機種コード、スーパーシート空席数、エコノミー前方窓側空席数、エコノミー前方通路側空席数、エコノミー後方窓側空席数、エコノミー後方通路側空席数及び特割残席数が含まれる。また、機種部３８１０には、キーである機種コード、属性である機種名、搭乗定員、航続距離、巡航速度及び飛行高度が含まれる。機種座席部３８２０には、キーである資源コード及び座席番号と、属性であるスーパーシートとエコノミーの別、窓側と通路側の別、前方と後方の別、及び喫煙と禁煙の別とが含まれる。座席部３８３０には、キーである出発日、便名及び座席番号と、属性である予約番号、登場お客様番号、搭乗者氏名、年齢性別、予約日時、発券日時、及び搭乗時刻が含まれる。なお、機種コードを含むようにしてもよい。予約部３８４０には、キーである予約番号、属性である予約お客様番号、予約者指名、代理店コード及び予約日時が含まれる。

【０１０４】

一般的に座席予約型データ構造には、図３９に示されるように、キーである機会キーと属性である資源グループ・キーとを含む機会部３９００と、キーである資源グループ・キーを含む資源グループ部３９１０と、キーである資源キーと属性である資源グループ・キーとを含む資源部３９２０と、キーである機会キー及び資源キーと属性である予約キーとを含むオカーレンス部３９３０と、キーである予約キーを含む予約部３９４０とが含まれる。

【０１０５】

機会部３９００と資源グループ部３９１０とはリンクしており、１つの資源グループ部３９１０に対して機会部３９００はＮ個（Ｎは０以上の整数）存在する。資源グループ部３９１０と資源部３９２０とはリンクしており、１つの資源グループ部３９１０に対して資源部３９２０はＭ個（Ｍは０以上の整数）存在する。資源部３９２０とオカーレンス部３９３０とはリンクしており、１つの資源部３９２０に対してオカーレンス部３９３０はＫ個（Ｋは０以上の整数）存在する。オカーレンス部３９３０と予約部３９４０とはリンクしており、１つの予約部３９４０に対してオカーレンス部３９３０はＬ個（Ｌは０以上の整数）存在する。

【０１０６】

このように、座席予約型データ構造のデータ構造部３２０に対する雛型プログラム３８０には、図３９に示されるようなデータ構造を図３８に示されるようなデータ構造にするような属性を与えるための解決ロジックが含まれる。また、機会部３９００、資源グループ部３９１０及び予約部３９４０の各々には、生成、削除、キー検索及び条件検索を実行する処理部が設けられている。さらに、オカーレンス部３９３０には、生成、削除及びキー検索を実行する処理部が設けられている。さらに、資源部３９２０には、キー検索及び条件検索を実行する処理部が設けられている。

【０１０７】

また、座席予約型データ構造の操作基本部３１０には、以下に示すような基本操作が含まれる。なお括弧内の文字列はメソッド名である。

（１）予約新規作成（キーあり）（返却値は真偽）（createSeatReservationWithKey1）

入力情報に従い、予約とオカーレンスを創成する。

（２）予約登録（キーあり）（返却値は情報）

入力情報に従い、予約とオカーレンスを創成する（createSeatReservationWithKey2）。そして、創生した予約エンティティの情報を返す。

（３）予約登録（キーなし）（返却値は情報）

入力情報に従い、予約とオカーレンスを創成する（createSeatReservationWithoutKey）。プライマリキーは採番クラスから取得して使用する。そして、創生した予約エンティティの情報を返す。

（４）予約取消（cancelSeatReservation）

入力情報の予約キーの予約を削除し、関連するオカーレンスを削除する。

（５）予約更新

入力情報に従い、予約を更新する（返却値は真偽）。

（６）予約更新

入力情報に従い、予約を更新する（返却値は情報）。

（７）予約キー検索（findSeatReservationByKey）

指定された予約キーの予約を取得し、その情報を出力用メッセージキャリアに

格納する。

（８）オカーレンス指定予約検索（findSeatReservationByOccurrence）

指定されたオカーレンスキーで予約を取得し、その情報を出力用メッセージキャリアに格納する。

（９）機会指定予約検索

指定された機会キーで予約を検索し、存在するかどうかを返す（findSeatReservationByOccasion）。又は、予約の複数検索の結果を指定されたメッセージキャリアに格納して返す（nextSeatReservation）。

（１０）資源指定予約検索

指定された資源キーで予約を検索し、存在するかどうかを返す（findSeatReservationByResource）。又は、予約の複数検索の結果を指定されたメッセージキャリアに格納して返す（nextSeatReservation）。

（１１）予約条件検索

指定された条件で予約を検索し、存在するかどうかを返す。又は、予約の複数検索の結果を指定されたメッセージキャリアに格納して返す（nextSeatReservation）。

（１２）予約へのオカーレンス追加（キーあり）（返却値は真偽）（createOccurrence1）

入力情報に従い、予約に対するオカーレンスを追加する。

（１３）予約へのオカーレンス追加（キーあり）（返却値は情報）

入力情報に従い、予約に対するオカーレンスを追加する（createOccurrence2）。又は、創生したオカーレンスエンティティの情報を返す。

（１４）オカーレンス削除（removeOccurrence）

入力情報に従い、予約に対するオカーレンスを削除する。

（１５）オカーレンス更新

入力情報に従い、オカーレンスを更新する（返却値は真偽）。

（１６）オカーレンス更新

入力情報に従い、オカーレンスを更新する（返却値は情報）。

（１７）オカーレンスキー検索（findOccurrenceByKey）

指定されたオカーレンスキーのオカーレンスを取得し、指定されたメッセージキャリアに格納する。

（１８）空き状況確認（isVacant）

指定された資源キーと機会キーに対応するオカーレンスが存在するか確認する（返却値は真偽）。

（１９）予約への機会変更（changeReservedOccasion）

指定された予約キーで予約を取得し、入力情報に指定された機会に変更する（返却値は真偽）。

（２０）予約への資源変更（changeReservedResource）

指定された予約キーで予約を取得し、入力情報に指定された資源に変更する（返却値は真偽）。

（２１）予約済オカーレンス件数検索

指定された予約キーでオカーレンスを検索し、件数を返す。

（２２）予約指定予約済オカーレンス検索

指定された予約キーでオカーレンスを検索し、存在するかどうかを返す（findOccurrenceBySeatReservation）。又は、オカーレンスの複数検索の結果を指定されたメッセージキャリアに格納して返す（nextOccurrence）。

（２３）機会指定予約済オカーレンス検索

指定された機会キーでオカーレンスを検索し、存在するかどうかを返す（findOccurrenceByOccasion）。オカーレンスの複数検索の結果を指定されたメッセージキャリアに格納して返す（nextOccurrence）。

【０１０８】

（２４）資源指定予約済オカーレンス検索

指定された資源キーでオカーレンスを検索し、存在するかどうかを返す（findOccurrenceByResource）。又は、オカーレンスの複数検索の結果を指定されたメッセージキャリアに格納して返す（nextOccurrence）。

（２５）オカーレンス条件検索

指定された条件でオカーレンスを検索し、存在するかどうかを返す。又は、オカーレンスの複数検索の結果を指定されたメッセージキャリアに格納して返す（

nextOccurrence）。

（２６）機会追加（キーあり）（返却値は真偽）（createOccasionWithKey1）

入力情報に従い、機会を追加する。

（２７）機会追加（キーなし）（返却値は情報）

入力情報に従い、機会を追加する（createOccasionWithoutKey）。プライマリキーは採番クラスから取得して使用する。そして、創生した機会エンティティの情報を返す。

（２８）機会削除（removeOccasion）

入力情報の機会キーの機会を削除する。機会に関連する予約済みオカーレンスが存在する場合は削除できない。

（２９）機会更新

入力情報に従い、機会を更新する（返却値は真偽）。

（３０）機会キー検索（findOccasionByKey）

指定された機会キーの機会を取得し、指定されたメッセージキャリアに格納する。

（３１）機会条件検索

指定された条件で機会を取得し、指定されたメッセージキャリアに格納する。そして、機会の複数検索の結果を指定されたメッセージキャリアに格納して返す（nextOccasion）。

（３２）資源グループ指定機会検索

指定された資源グループキーで機会を検索し、存在するかどうかを返す。又は、機会の複数検索の結果を指定されたメッセージキャリアに格納して返す（nextOccasion）。

（３３）資源指定機会検索

指定された資源キーで機会を検索し、存在するかどうかを返す（findOccasionByResource）。機会の複数検索の結果を指定されたメッセージキャリアに格納して返す（nextOccasion）。

（３４）資源指定空き機会検索

指定された資源キーで空き機会を検索し、存在するかどうかを返す（findVaca

ntOccasionByResource）。機会の複数検索の結果を指定されたメッセージキャリアに格納して返す（nextOccasion）。

（３５）空き機会条件検索

指定された条件で空き機会を検索し、存在するかどうかを返す。又は、機会の複数検索の結果を指定されたメッセージキャリアに格納して返す（nextOccasion）。

（３６）資源キー検索（findOccasionByKey）

指定された資源キーの資源を取得し、指定されたメッセージキャリアに格納する。

（３７）資源条件検索

指定された条件で資源を検索し、存在するかどうかを返す。又は、資源の複数検索の結果を指定されたメッセージキャリアに格納して返す（nextResource）。

（３８）機会指定資源検索

指定された機会キーで資源を検索し、存在するかどうかを返す。又は、資源の複数検索の結果を指定されたメッセージキャリアに格納して返す（nextResource）。

（３９）資源グループ指定資源検索

指定された資源グループキーで資源を検索し、存在するかどうかを返す。資源の複数検索の結果を指定されたメッセージキャリアに格納して返す（nextResource）。

（４０）機会指定空き資源検索

指定された機会キーで空き資源を検索し、存在するかどうかを返す（findVacantResourceByOccasion）。又は、資源の複数検索の結果を指定されたメッセージキャリアに格納して返す（nextResource）。

（４１）資源グループ追加（キーあり、返却値は真偽）

入力情報に従って資源グループを追加する（返却値は真偽）。

（４２）資源グループ削除

指定された資源グループキーの資源グループを削除（物理削除）する。

（４３）資源グループ更新

入力情報に従い、資源グループを更新する（返却値は真偽）。

（４４）資源グループキー検索

指定された資源グループキーの資源グループを取得し、情報を出力用メッセージキャリアに格納する。

（４５）資源グループ条件検索

指定された条件に合う資源グループを検索し、存在するかどうかを返す。又は、資源グループの複数検索の結果を指定されたメッセージキャリアに格納して返す（nextResourceGroup）。

このような座席予約型データ構造に対する基本操作に対応する各雛型プログラムには、対象となるプログラム固有の設定を行うための解決ロジックが埋め込まれている。

【０１０９】

１０．系図型データ構造の場合

系図型データ構造は、職制変更で起きる組織の統廃合などのように、作成、廃止、統合などに伴う責任範囲の引き継ぎ先や引き継ぎ元の情報を管理するために用いられる。

【０１１０】

図４０を用いて系図型データ構造の適用可能例を示す。図４０の例では、（１）という組織が新規に作成され、その後（４）という組織が（１）から分割して作成された。（４）にとって（１）は相続元である。また、（２）という組織が新規に作成され、（１）は（２）に吸収されて消滅した。よって、（２）は（１）の相続先である。一方、（３）という組織は新規に作成され、相続先がなく消滅した。このように、（１）乃至（４）のような系ノードのインスタンスは、自身の相続元と相続先へのリンクをそれぞれ１つずつ持つことができる。また、相続元を持たないインスタンスは、それ自身が起源となる。さらに、相続先を持たないインスタンスは実世界上存在していることを示し、相続先を持っているインスタンスは実世界上消滅していることを示している。

【０１１１】

同じ相続元を持つインスタンスが複数存在する場合はあっても、相続元から見

ると相続先は一つである。また、第１のインスタンスの相続先である第２のインスタンスは、必ずしも第１のインスタンスが相続元であるというデータを有していない場合もある。第２のインスタンスが他のインスタンスから相続し、且つ第１のインスタンスが勝手に相続先であると指定する場合である。

【０１１２】

このような性質を実現するための系図型データ構造は、図４１に示すように、系ノード部４１００を含む。系ノード部４１００には、キーである系ノードと、属性である相続元系ノードキー、相続先系ノードキー、発生日、消滅日、及び属性１乃至ｎが含まれる。同じ相続元を持つインスタンスが複数存在する場合でも、その相続元から見た相続先は１つだけである。また、あるインスタンスが持つ相続先が、逆にその相続元として必ずしも自分を指していなくとも良い。なお、系図型データ構造のデータ構造部３２０に対応する雛型プログラム３８０には、系ノード部４１００に属性を与えるための解決ロジックが含まれる。また、系ノード部４１００には、生成、削除、キー検索、吸収ノード検索及び分裂ノード検索を実行する処理部が設けられている。

【０１１３】

また、系図型データ構造の操作基本部３１０には、以下に示すような基本操作が含まれる。なお括弧内の文字列はメソッド名である。

（１）系ノードキー検索（findNodeByKey）

系ノードのキーを指定して、系ノードを検索する。

（２）相続元検索（findParentOfNode）

系ノードのキーを指定して、その系ノードの相続元ノードを検索する。

（３）相続先検索（findChildOfNode）

系ノードのキーを指定して、その系ノードの相続先ノードを検索する。

（４）吸収ノード検索

系ノードのキーを指定して、その系ノードを相続先とする系ノードを検索する（findAbsorbedNodesOfNodes）。そして、検索されたノードリストから、ノード情報を取り出して、返却値を作る（nextNode）。

（５）分裂ノード検索

系ノードのキーを指定して、その系ノードを相続元とする系ノードを検索する（findDividedNodesOfNode）。そして、検索されたノードリストから、ノード情報を取り出して、返却値を作る（nextNode）。

（６）起源検索

系ノードをキーを指定して、その系ノードの起源となる相続元ノードを検索する（findRootOfNode）。

（７）最終相続先検索（findLeafOfNode）

系ノードをキーを指定して、その系ノードの最終相続先ノードを検索する。

（８）同一系チェック（上方向）（isAccessibleAncestor）

起点となる系ノードから、ある系ノードが上方向にアクセス可能か調査する。

（９）同一系チェック（下方向）（isAccessibleDescendant）

起点となる系ノードから、ある系ノードが下方向にアクセス可能か調査する。

（１０）条件検索

ノードを条件指定で検索する。

（１１）キー付きで系ノード生成（返却値は真偽）（createNodeWithKey1）

キーを指定して、系ノードを新規に生成する。

（１２）キー付きで系ノード生成（返却値は生成したノードの情報）（createNodeWithKey2）

キーを指定して、系ノードを新規に生成する。

（１３）キー無しで系ノード生成（返却値は生成したノードの情報）（createNodeWithoutKey）

キーを指定しないで、系ノードを新規に生成する。キーは採番サービスから取得する。

（１４）系ノードの削除（removeNode）

キーで指定した系ノードを削除する。他の系ノードからリンクを張られている場合は削除不可。

（１５）系ノードを消滅状態にする（dissolveNode）

系ノードの消滅日に日付を設定し、系ノードを消滅状態にする。

（１６）消滅状態の系ノードを復活する（reviveNode）

系ノードの消滅日をnullに設定し、消滅状態を無効にする。

（１７）ノード情報の更新

系ノードの情報を更新する。返却値は真偽の場合と、ノードの情報の場合がある。

（１８）系ノードの転換

系ノードを生成し、相続元ノードは消滅状態とする。

（１９）系ノードの吸収

複数の系ノードの相続先を既存の１つの系ノードに設定する。

（２０）系ノードの合併

複数の系ノードの相続先を新規に生成した１つの系ノードに設定する。

（２１）系ノードの分裂

系ノードを指定し、その系ノードを相続元とする複数の系ノードを新規に生成する。

（２２）系ノードの分割

系ノードを指定し消滅させ、その系ノードを相続元とする複数の系ノードを新規に生成する。

このような系図型データ構造に対する基本操作に対応する各雛型プログラムには、対象となるプログラム固有の設定を行うための解決ロジックが埋め込まれている。

【０１１４】

１１．在庫型データ構造の場合

在庫型データ構造は、在庫や残高など、何らかのトランザクションを受けた状態変化を管理するときに使用される。在庫型データ構造は、図４２に示すように、現在倉庫等に存在している在庫のための在庫部４２００及び引当明細部４２１０と、将来倉庫等に入れられる未来在庫のための入荷予定部４２２０及び入荷予定引当明細部４２３０とを含む。在庫部４２００に対して引当明細部４２１０がＮ個（Ｎは０以上の整数）存在する。入荷予定部４２２０に対して入荷予定引当明細部４２３０がＮ個（Ｎは０以上の整数）存在する。

【０１１５】

在庫部４２００は、キーである在庫キー（「倉庫コード（倉庫が複数ある場合に必要）、商品コード、所有者区分、所有者コード、規格（色柄・サイズ）コード、ロット番号」等の組合せ）と、属性である引当可能数及び引当総数とが含まれる。引当可能数（未引当の数）＋引当総数（引当られた数）＝在庫数となる。引当明細部４２１０は、キーである在庫キー及び引当明細キーと、属性である引当要求数（予定数）、引当数（実際に引き当てられた数）、出庫済数（実際に出庫した数）及び出庫予定日とが含まれる。引当明細部４２１０は商品等を受注すると生成される。但し、初めから引当明細部４２１０が設けられているようにすることも可能である。

【０１１６】

入荷予定部４２２０は、キーである入荷予定キー（未定部分もあるので在庫キーより簡単な場合もある）と、属性である入荷予定日、入荷予定数、入荷実績数、引当可能数、及び引当総数とを含む。また、入荷予定引当明細部４２３０は、キーである入荷予定キー及び引当明細キーと、属性である引当要求数及び引当数とを含む。

【０１１７】

なお、在庫型データ構造のデータ構造部３２０に対応する雛型プログラム３８０には、在庫部４２００、引当明細部４２１０、入荷予定部４２２０、及び入荷予定引当明細部４２３０に属性を与えるための解決ロジックが含まれる。また、在庫部４２００、引当明細部４２１０、入荷予定部４２２０、及び入荷予定引当明細部４２３０の各々には、生成、削除、キー検索及び条件検索を実行する処理部が設けられている。

【０１１８】

また、在庫型データ構造の操作基本部３１０には、以下に示すような基本操作が含まれる。なお括弧内の文字列はメソッド名である。

（１）在庫検索（findStockByKey）

指定されたキーを持つ在庫エンティティのインスタンスを検索する。

（２）在庫条件検索

条件に適合する在庫エンティティのインスタンスの集合を準備し、その準備が

できたかどうかを返す。そして、次の在庫エンティティの情報を取得する（nextStock）。

（３）在庫登録（createStockWithKey）

指定されたキーを持つ在庫エンティティのインスタンスを生成する（返却値は真偽）。

（４）在庫登録（返却値はエンティティの情報）

指定されたキーを持つ在庫エンティティのインスタンスを生成する（createStockWithKey）。そして、生成した在庫エンティティの情報を返す。

（５）引当可能数設定（setStockAvailableQuantity）

指定されたキーを持つ在庫エンティティのインスタンスを検索し、指定された値を抽象レベルの属性である引当可能数に設定する。

（６）在庫削除（removeStock）

指定されたキーを持つ在庫エンティティのインスタンスを削除する。

（７）在庫修正

指定されたキーを持つ在庫エンティティのインスタンスの情報を更新する（返却値は真偽）。

（８）在庫修正

指定されたキーを持つ在庫エンティティのインスタンスの情報を更新する（返却値は更新したエンティティの情報）。

（９）在庫引当検索（findReservationByKey）

指定されたキーを持つ在庫引当明細エンティティのインスタンスを検索する。

（１０）在庫全引当検索

指定された在庫キーを持つ在庫引当明細エンティティのインスタンスを全て検索する（findReservationsByStock）。そして、次の在庫引当明細エンティティの情報を取得する（nextReservation）。

（１１）在庫引当（createReservation）

指定された在庫を引き当て、在庫引当明細エンティティのインスタンスを１件生成する（返却値は生成したエンティティの情報）。そして、生成した在庫引当明細エンティティの情報を返す。

（１２）複数件在庫引当（createReservations）

指定された複数件の在庫を引き当て、在庫引当明細エンティティのインスタンスをそれぞれ１件ずつ生成する（返却値は指定した在庫の件数）。

（１３）条件指定在庫引当

条件に適合する在庫を引き当て、在庫引当明細エンティティのインスタンスを生成する（返却値は生成した在庫引当明細エンティティの情報）。

（１４）在庫引当戻し（removeReservation）

指定されたキーを持つ在庫引当明細エンティティのインスタンスを削除する。つまり、在庫引当のキャンセルである。同時に、在庫の引当総数を減らし、引当可能数を増やす。ただし、在庫引当に出庫数が入っている場合には、引当戻しはできない。

（１５）在庫引当修正

指定されたキーを持つ在庫引当明細エンティティのインスタンスの情報を更新する（返却値は真偽）。

（１６）在庫引当修正

指定されたキーを持つ在庫引当明細エンティティのインスタンスの情報を更新する（返却値は更新したエンティティの情報）。

（１７）在庫引落（引当ありの引落）

引当ありの引落であり、指定されたキーを持つ在庫引当明細エンティティのインスタンスに対し、引き落としを行ない、出庫数を設定する（返却値は在庫エンティティの情報）（deliverWithReservation）。そして、引き落とされた在庫引当明細エンティティに対応する在庫エンティティの情報を返す。

（１８）在庫引落（引当なしの引落）

引当なしの引落であり、指定されたキーを持つ在庫エンティティのインスタンスに対し、引き落としを行ない、出庫数を設定する。同時に、引き落とされた在庫に対応する在庫引当明細エンティティのインスタンスを生成する（返却値は更新した在庫エンティティの情報）（deliverWithoutReservation）。そして、更新した在庫エンティティの情報を返す。

（１９）出荷（shipReservation）

引き当てた在庫を出荷する。指定されたキーを持つ在庫引当明細エンティティのインスタンスを削除し、同時に在庫の引当総数を消し込む（返却値は在庫エンティティの情報）。

（２０）入荷予定検索（findFutureReceptionByKey）

指定されたキーを持つ入荷予定エンティティのインスタンスを検索する。

（２１）入荷予定条件検索

条件に適合する入荷予定エンティティのインスタンスの集合を準備し、その準備ができたかどうかを返す。又は、次の入荷予定エンティティの情報を取得する（nextFutureReception）。

（２２）入荷予定登録（createFutureReceptionWithKey）

指定されたキーを持つ入荷予定エンティティのインスタンスを生成する（返却値は真偽）。

（２３）入荷予定取消し（removeFutureReception）

指定されたキーを持つ入荷予定エンティティのインスタンスを削除する。すなわち、入荷予定のキャンセルである。引当情報の有無にかかわらず入荷予定をキャンセルにする。

（２４）入荷予定修正

指定されたキーを持つ入荷予定エンティティのインスタンスの情報を更新する（返却値は真偽）。

（２５）入荷予定修正

指定されたキーを持つ入荷予定エンティティのインスタンスの情報を更新する（返却値は更新したエンティティの情報）。

（２６）入荷予定引当検索（findFutureReservationByKey）

指定されたキーを持つ入荷予定引当明細エンティティのインスタンスを検索する。

（２７）入荷予定全引当検索

指定された入荷予定キーを持つ入荷予定引当明細エンティティのインスタンスを全て検索する（findFutureReservationsByFutureReception）。又は、次の入荷予定引当明細エンティティの情報を取得する（nextFutureReservation）。

（２８）入荷予定引当

指定された入荷予定を引き当て、入荷予定引当明細エンティティのインスタンスを生成する（返却値は生成した入荷予定引当明細エンティティの情報）（createFutureReservation）。そして、生成した入荷予定引当明細エンティティの情報を返す。

（２９）入荷予定引当取消し（removeFutureReservation）

入荷予定引当明細エンティティのインスタンスを削除する。つまり、入荷予定引当のキャンセルである。

（３０）入荷予定引当修正

指定されたキーを持つ入荷予定引当明細エンティティのインスタンスの情報を更新する（返却値は真偽）。

（３１）入荷予定引当修正

指定されたキーを持つ入荷予定引当明細エンティティのインスタンスの情報を更新する（返却値は更新したエンティティの情報）。

このような在庫型データ構造に対する基本操作に対応する各雛型プログラムには、対象となるプログラム固有の設定を行うための解決ロジックが埋め込まれている。

【０１１９】

１２．計画型データ構造の場合、

計画型データ構造は、販売計画、生産計画などの計画値とその実績値を管理する時に使われる。計画業務においては、時間の経過により計画が進行し、例えば製品毎や組織毎の売上目標及び実績などが管理される。よって、時間という軸と、製品、組織といった他の軸とが定義され、時間の経過に従った製品毎の目標及び実績、時間の経過に従ったある組織毎の目標及び実績、時間の経過に従った製品及び組織毎の目標及び実績といった各軸の組合せに係るデータが管理される。なお、軸の種類は、時間のみ決まっており他の軸の種類は任意である。数も任意である。

【０１２０】

例えば図４３に示すように、製品（Ｘ軸）、組織（Ｙ軸）、時間（Ｔ軸）が規

定され、製品の階層を表す製品種、機種、製品それぞれについて、組織の階層を表す本部、事業部、部についての目標値及び実績値が管理される。また、時間軸も、年度についてのデータ、月度についてのデータ、日次についてのデータが管理される。但し、図４３の時間（Ｔ軸）の列で○が付された部分だけ、実際にはデータの管理がなされる。すなわち、製品種については、本部が年度で、事業部が年度と月度でデータを管理する。機種については、事業部が年度及び月度、部が年度及び月度でデータを管理する。製品については、事業部が月度で、部が年度、月度及び日次でデータを管理する。

【０１２１】

上で述べたような業務に対応すべく計画型データ構造は、図４４に示すような構造を有している。すなわち、計画部４４００と、時間軸部４４１０と、時間軸階層部４４２０と、軸種別部４４３０とが含まれる。なお、軸種別部４４３０が存在するのは、当該軸が表すデータが単純なデータでなく、階層構造を有している場合（上で示した製品の場合（製品に関するコードが階層化されている場合））等のために設けられる。ここではエンティティとして示さないが、Ｙ軸のためのエンティティを用意しても良いし、他のシステムから取り込むような形（ビューを用意する）にしても良い。例えば、Ｙ軸に対応するデータが、単純型のデータ構造の場合やツリー型のデータ構造の場合（上で示した組織の場合）には、ビューを設けるようにしてもよい。ここでビューとは、参照のみ可能なエンティティのことである。

【０１２２】

この計画部４４００には、時間軸を特定するための時間軸コード（キー）と、製品種等を表すＸ軸の種別を特定するためのＸ軸種別（キー）と、Ｘ軸の内容を表すＸキー（キー）と、Ｙ軸の内容を表すＹキー（キー）と、計画のバージョンを表す版数（キー）と、属性である予定値及び実績値とが含まれる。例えば、時間軸コードとして「１９９９０４」（９９年４月度）、Ｘ軸種別として「製品種」、Ｘキーとして「ＰＣ」（製品種コード）、Ｙキーとして「２３４」（事業部コード）、版数として「０１」、予定値として「１２０００００」、実績値として「１０３２１０００」が設定される。

【０１２３】

時間軸部４４１０には、時間軸コード（キー）と、属性である時間軸名及び開始日付とが含まれる。時間軸部４４１０は、年度、半期、四半期、月度といった時間にも種別があり、且つ開始日付が異なるために設けられている。例えば、時間軸コードとして「１９９９１１」、時間軸名として「１９９９年１１月度」、開始日付として「１９９９１０２１」というデータが設定される。

【０１２４】

時間軸階層部４４２０には、時間軸階層コード（キー）と、属性である時間軸階層名、上位時間軸階層コード、及び下位時間軸階層コードとが含まれる。時間軸階層部４４２０は、時間が例えば年度、月度、日次といったように階層的に管理されるために設けられている。上位時間軸階層コード及び下位時間軸階層コードが設定され、ここでは上下関係が一対一で決定される。例えば、時間軸階層コードとして「ＧＡＴＳＵＤＯ」、時間軸階層名として「月度」、上位時間軸階層コードとして「ＮＥＮＤＯ」、下位時間軸階層コードとして「ＮＩＣＨＩ」というデータが設定される。軸種別部４４３０には、Ｘ軸種別（キー）が含まれる。

【０１２５】

このように、計画型データ構造のデータ構造部３２０に対応する雛型プログラム３８０には、計画部４４００、時間軸部４４１０、時間軸階層部４４２０、及び軸種別部４４３０に属性を与えるための解決ロジックが含まれる。また、少なくとも計画部４４００及び時間軸部４４１０には、生成、削除、キー検索及び条件検索を実行する処理部が設けられる。

【０１２６】

また、計画型データ構造の操作基本部３１０には、以下に示すような基本操作が含まれる。なお括弧内の文字列はメソッド名である。

（１）時間軸登録（createTimeAxis1）

時間軸を新規に登録する。返却値は真偽値。エラーが起きた場合、偽（false）を返す。

（２）時間軸登録（createTimeAxis2）

時間軸を新規に登録する。返却値は生成した時間軸の情報。

（３）時間軸一括登録（createTimeAxisTree）

ツリー構造の時間軸を登録する。下階層の時間軸は上位階層の期間内で、かつ、開始日が一致しなければならない。

（４）時間軸削除（removeTimeAxisTree）

指定された時間軸とその下位の時間軸を削除する。ただし、時間軸とリンクしている計画が一つでもある場合は、これらの計画のキーを返し、時間軸も削除しない。

（５）時間軸更新（updateTimeAxis1）

指定した時間軸の情報を更新する（返却値は真偽値）。エラーが発生した場合は、falseを返す。

（６）時間軸更新（updateTimeAxis2）

指定した時間軸の情報を更新する。返却値は更新した時間軸の情報。

（７）時間軸キー検索（findTimeAxisByKey）

指定したキーに該当する時間軸を検索する。該当する時間軸の情報を返す。

（８）時間軸使用チェック（checkUsedTimeAxisTree）

指定した時間軸を参照する計画を検索する。該当する計画の集合を返す。

（９）期間指定による子時間軸変換（convertTimeAxisKeys）

開始日付と時間軸階層を指定して、その期間に属する子時間軸を返す。

（１０）計画予定登録（createPlan1）

計画予定を新規に登録する（返却値は真偽）。エラーが発生した場合は、falseを返す。

（１１）計画予定登録（createPlan2）

計画予定を新規に登録する（返却値は情報）。登録した計画予定の情報を返す。

（１２）計画実績登録（createResult1）

計画実績を新規に登録する（返却値は真偽値）。エラーが発生した場合、falseを返す。

（１３）計画実績登録（createResult2）

計画実績を新規に登録する。登録した計画実績の情報を返す。

（１４）時間軸生成付き計画予定登録（createPlanWithAxis）

指定された計画予定と時間軸の両方を生成する。生成した計画予定の情報を返す。

（１５）時間軸生成付き計画実績登録（createResultWithAxis）

指定された計画実績と時間軸の両方を生成する。生成した計画実績の情報が返す。

（１６）計画予定更新（updataPlan）

指定された計画予定の内容を変更する。エラーがあれば、falseが返る。

（１７）計画実績更新（updateResult）

指定された計画実績の内容を変更する。エラーがあれば、falseが返る。

（１８）計画予定値更新（updatePlanValue）

指定された計画予定の予定値を変更する。エラーがなければ、更新後の計画予定の情報が返る。

（１９）計画実績値更新（updateResultValue）

指定された計画実績の実績値を変更する。エラーがなければ、更新後の計画実績の情報が返る。

（２０）計画予定キー検索（findPlanByKey）

指定したキーに該当する計画予定を検索する。該当する計画予定の情報を返す。

（２１）計画実績キー検索（findResultByKey）

指定したキーに該当する計画実績を検索する。該当する計画実績の情報を返す。

（２２）軸指定一括計画予定検索（findPlansByAxis）

この検索機能では、入力として検索する計画予定の情報と取得したい直下の計画予定の軸を指定する。この機能を実行すると、(1)検索で該当した計画予定の情報と、(2)指定軸に関してその（該当した計画予定の）直下に存在する計画予定の全てが得られる。

（２３）軸指定一括計画実績検索（findResultsByaxis）

この検索機能では、入力として検索する計画実績の情報と取得したい直下の計

画実績の軸を指定する。この機能を実行すると、(1)検索で該当した計画実績の情報と、(2)指定軸に関してその（該当した計画実績の）直下に存在する計画実績の全てが得られる。

（２４）計画予定削除（removePlan）

指定した計画予定を削除する。

（２５）計画実績削除（removeResult）

指定した計画実績を削除する。

（２６）予定集約値算出（calcPlanValue）

指定した計画予定の予定値の合計を、指定した軸に関してその（指定した計画予定の）直下に存在する計画予定の予定値を加算して求める。

（２７）実績集約値算出（calcResultValue）

指定した計画実績の実績値の合計を、指定した軸に関してその（指定した計画実績の）直下に存在する計画実績の実績値を加算して求める。

（２８）予定消化率算出（calcPlanUsedRate）

指定した計画予定の予定値に対して、指定した軸でその（指定した計画予定の）直下に存在する計画予定の予定値の合計が占める（消費した）割合を求める。

（２９）予実差異算出（calcDifference）

指定された計画予定の予定値と計画実績の実績値の差異を計算する。

このような計画型データ構造に対する基本操作に対応する各雛型プログラムには、対象となるプログラム固有の設定を行うための解決ロジックが埋め込まれている。

【０１２７】

上で述べた実施の形態は、一例であって様々な変形が可能である。例えば、ユーザインターフェースの例を図示しているが、同様の内容の他の画面であってもよい。また、各データ構造について基本操作の例を示しているが、基本操作は上記のとおりでなくともよく、より少ない基本操作を提供して残りはユーザが別途用意するようにしてもよい。また、より多くの基本操作を提供してユーザが別途用意しなくとも良くするような態様であってもよい。

【０１２８】

データ構造は一例であって、他のデータ構造を規定してもよいし、より少ない種類のデータ構造を実装する場合もある。

【０１２９】

また、対象となるプログラムを生成する本プログラム自動生成装置は、プログラムとコンピュータの組合せにて実現する例を上で示したが、専用の回路などを組み合わせて実現することも可能である。さらに、プログラムとコンピュータの組合せにて実現する場合、当該プログラムをフロッピー・ディスク、ＣＤ－ＲＯＭ、ＤＶＤ、ＨＤＤ、半導体メモリなどの記憶媒体又は記憶装置に格納することも可能である。

【０１３０】

（付記１）

所定の処理を行うプログラムを自動的に生成するプログラム自動生成装置であって、

前記所定の処理固有の設定を行うための解決ロジックを含み且つ予め対応付けられたデータ構造のための雛型プログラムを各々含む、複数のデータ構造解決ユニットと、

選択されたデータ構造に対応する前記データ構造解決ユニット内の前記雛型プログラムに含まれる前記解決ロジックの、前記所定の処理固有の設定に関する解決情報を取得し、当該解決ロジックの解決情報と前記雛型プログラムとを合成することにより、前記所定の処理を行うプログラムを生成する解決器と、

を有するプログラム自動生成装置。

【０１３１】

（付記２）

前記解決器が、

前記選択されたデータ構造に対応する前記データ構造解決ユニット内の前記雛型プログラムに含まれる前記解決ロジックを解析し、当該解決ロジックに対する解決情報の入力をユーザに促す手段

を有することを特徴とする付記１記載のプログラム自動生成装置。

【０１３２】

（付記３）

前記選択されたデータ構造に対応する前記データ構造解決ユニットは、単純型データ構造に対応するデータ構造解決ユニット、伝票型データ構造に対応するデータ構造解決ユニット、階層型データ構造に対応するデータ構造解決ユニット、ツリー型データ構造に対応するデータ構造解決ユニット、在庫型データ構造に対応するデータ構造解決ユニット、時間帯予約型データ構造に対応するデータ構造解決ユニット、計画型データ構造に対応するデータ構造解決ユニット、座席予約型データ構造に対応するデータ構造解決ユニット、構成型データ構造に対応するデータ構造解決ユニット、明細主導伝票型データ構造に対応するデータ構造解決ユニット、系図型データ構造に対応するデータ構造解決ユニット、及びマトリックス型データ構造に対応するデータ構造解決ユニットのうちいずれかのデータ構造解決ユニットであることを特徴とする付記１記載のプログラム自動生成装置。

【０１３３】

（付記４）

前記データ構造解決ユニットが、

１又は複数のレコードタイプと、前記複数のレコードタイプが存在する場合には当該複数のレコードタイプ間のリンクとから構成されるデータ構造を規定し、当該所定のデータ構造についての前記所定の処理のための設定を行うための解決ロジックを含む第１の雛型プログラムと、

操作について前記所定の処理のための設定を行うための解決ロジックを含み、前記所定のデータ構造に対して実行される基本操作に対応する第２の雛型プログラムと、

を含むことを特徴とする付記１記載のプログラム自動生成装置。

【０１３４】

（付記５）

所定の処理を行うプログラムを自動的に生成するプログラム自動生成プログラムを格納する記録媒体であって、

前記プログラム自動生成プログラムは、

前記所定の処理固有の設定を行うための解決ロジックを含み且つ予め対応付け

られたデータ構造のための雛型プログラムを含むデータ構造解決ユニットにおいて、前記雛型プログラムに含まれる前記解決ロジックの、前記所定の処理固有の設定に関する解決情報を取得し、当該解決ロジックの解決情報と前記雛型プログラムとを合成することにより、前記所定の処理を行うプログラムを生成する解決プログラム、

を含む記録媒体。

【０１３５】

（付記６）

所定の仕様に従った第１プログラムを生成するために使用される第２プログラムを格納する記録媒体であって、

前記第２プログラムは、

１又は複数のレコードタイプと、前記複数のレコードタイプが存在する場合には当該複数のレコードタイプ間のリンクとから構成されるデータ構造を規定し、当該所定のデータ構造についての前記所定の仕様に従った設定を行うための解決ロジックを含む第１の雛型プログラムと、

操作について前記所定の仕様に従った設定を行うための解決ロジックを含み、前記所定のデータ構造に対して実行される基本操作に対応する第２の雛型プログラムと、

を含む記録媒体。

【０１３６】

（付記７）

前記第１の雛型プログラムが、１種類のレコードタイプからなるデータ構造を規定し、当該レコードの属性を与えるための解決ロジックを含み、

前記第２の雛型プログラムが、前記所定の仕様に従った設定を行うための解決ロジックが埋め込まれており、前記レコードタイプに対し、少なくとも追加操作、削除操作、更新操作、及び検索操作を実行するためのものである、

ことを特徴とする付記６記載の記録媒体。

【０１３７】

（付記８）

前記第１の雛型プログラムが、１種類のヘッダ・レコードタイプと１種類の明細レコードタイプと１つの前記ヘッダ・レコードタイプに対し１又は複数の前記明細レコードタイプを結ぶリンクとを有するデータ構造を規定し、前記ヘッダレコード及び前記明細レコードの属性を与えるための解決ロジックを含み、

前記第２のプログラムが、前記データ構造に対して少なくとも伝票新規作成及び伝票検索を実行するための第２の雛型プログラムであって、前記ヘッダ・レコードの状態を操作との関連で定義するための解決ロジックと、前記所定の仕様に従った設定をレコードの属性、レコードの状態、若しくはレコードの属性及び状態の組合せで記述するための解決ロジックとが埋め込まれている、

ことを特徴とする付記６記載の記録媒体。

【０１３８】

（付記９）

前記第１の雛型プログラムが、資源レコードタイプと、資源の予約に関する予約レコードタイプと、予約の時間の刻みに関する予約セル・レコードタイプと、１つの資源レコートタイプと１又は複数の予約レコードタイプとを結ぶリンクと、１つの予約レコードタイプと１又は複数の予約セル・レコードタイプを結ぶリンクとから構成されるデータ構造を規定し、前記データ構造内の各レコードの属性を与えるための解決ロジックを含み、

前記第２の雛型プログラムが、前記所定の仕様に従った設定を行うための解決ロジックを含み、前記データ構造に対して少なくとも予約登録、予約削除、予約更新及び予約検索を実行するためのものである、

ことを特徴とする付記６記載の記録媒体。

【０１３９】

（付記１０）

前記第１の雛型プログラムが、行レコードタイプと、列レコードタイプと、行及び列の交点を表すセル・レコードタイプと、各セルに対して割り当て可能な属性を表す列タイプ・レコードタイプとを含むデータ構造を規定し、行レコード、列レコード及び列タイプ・レコードの属性を与える解決ロジック及びセルに対して列タイプ・レコードを割り当てるための解決ロジックとを含み、

前記第２の雛型プログラムが、前記所定の仕様に従った設定を行うための解決ロジックを含み、前記データ構造の行レコード、列レコード及び列タイプ・レコードに対して、少なくとも追加操作、変更操作、削除操作及び検索操作を実行するためのものである、

ことを特徴とする付記６記載の記録媒体。

【０１４０】

（付記１１）

前記第１の雛型プログラムが、複数種類のヘッダ・レコードタイプと、１種類の明細レコードタイプと、前記複数種類のヘッダ・レコードタイプと前記１種類の明細レコードタイプとを結ぶリンクとを有するデータ構造を規定し、前記データ構造内の各レコードの属性を与える解決ロジックを含み、

前記第２のプログラムが、前記データ構造に対し少なくとも伝票新規作成、伝票検索、伝票削除及び伝票ヘッダ種変更を実行するための第２の雛型プログラムであって、前記ヘッダ・レコードの状態を操作との関連で定義するための解決ロジックと、前記所定の仕様固有の設定をレコードの属性、レコードの状態、若しくはレコードの属性及び状態の組合せで記述するための解決ロジックが埋め込まれている、

ことを特徴とする付記６記載の記録媒体。

【０１４１】

（付記１２）

前記第１の雛型プログラムが、在庫レコードタイプと、在庫の引当を規定するための在庫引当明細レコードタイプと、入荷予定レコードタイプと、入荷予定に対する引当を規定するための入荷予定引当明細レコードタイプと、前記在庫レコードタイプと前記在庫引当明細レコードタイプとを結ぶリンクと、前記入荷予定レコードタイプと前記入荷予定引当明細レコードタイプとを結ぶリンクとを有するデータ構造を規定し、前記データ構造内の各レコードの属性を与える解決ロジックを含み、

前記第２のプログラムが、前記データ構造に含まれる各レコードに対し少なくとも追加操作、削除操作、及び検索操作を実行するための第２の雛型プログラム

であって、前記所定の仕様に従った設定を行うための解決ロジックを含む、

ことを特徴とする付記６記載の記録媒体。

【０１４２】

（付記１３）

前記第１の雛型プログラムが、資源レコードタイプと、資源のグループを規定するための資源グループ・レコードタイプと、資源グループを利用可能な機会を規定する機会レコード・タイプと、機会と資源の組み合わせを規定するためのオカーレンス・レコードタイプと、１又は複数のオカーレンスに関連する予約レコード・タイプとを含むデータ構造を規定し、前記データ構造内の各レコードの属性を与える解決ロジックを含み、

前記第２の雛型プログラムが、前記データ構造に含まれる各レコードに対し少なくとも生成操作、削除操作、及び検索操作を実行するための第２の雛型プログラムであって、前記所定の仕様に従った設定を行うための解決ロジックを含む、

ことを特徴とする付記６記載の記録媒体。

【０１４３】

（付記１４）

前記第１の雛型プログラムが、計画レコードタイプと、時間軸に関する時間軸レコードタイプと、前記時間軸の管理単位を規定する時間軸階層レコードタイプとを含むデータ構造を規定し、計画レコード、時間軸レコード及び時間軸階層レコードの属性を与える解決ロジックと、計画の管理単位を指定するための解決ロジックとを含み、

前記第２の雛型プログラムが、前記データ構造に含まれる各レコードに対し少なくとも追加操作、削除操作、及び検索操作を実行するための第２の雛型プログラムであって、前記所定の仕様に従った設定を行うための解決ロジックを含む、

ことを特徴とする付記６記載の記録媒体。

【０１４４】

（付記１５）

前記第１の雛型プログラムが、ノード・レコードタイプと、当該ノードの版数管理のためのノード版数レコードタイプと、ノード版数レコード間の構成関係を

表す構成レコードタイプとを含むデータ構造を規定し、前記データ構造に含まれる各レコードの属性を与える解決ロジックと、

前記第２の雛型プログラムが、前記データ構造に含まれる各レコードに対して少なくとも追加操作、削除操作、及び検索操作を実行するための第２の雛型プログラムであって、前記所定の仕様に従った設定を行うための解決ロジックが埋め込まれている、

ことを特徴とする付記６記載の記録媒体。

【０１４５】

（付記１６）

前記第１の雛型プログラムが、系の連続性を保持するための関連情報が付加された１種類のレコードタイプからなるデータ構造を規定し、当該レコードの属性を与えるための解決ロジックが含まれ、

前記第２の雛型プログラムが、前記レコードに対して少なくとも追加操作、削除操作、及び検索操作を実行するための第２の雛型プログラムであって、前記所定の仕様に従った設定を行うための解決ロジックを含む、

ことを特徴とする付記６記載の記録媒体。

【０１４６】

（付記１７）

前記第１の雛型プログラムが、上位のノードに関する情報を保持するレコードタイプからなるデータ構造を規定し、当該レコードの属性を与えるための解決ロジックを含み、

前記第２の雛型プログラムが、前記レコードに対して少なくとも追加操作、削除操作、及び検索操作を実行するための第２の雛型プログラムであって、前記所定の仕様に従った設定をレコードの属性、レコードの状態、若しくはレコードの属性及び状態の組合せで記述するための解決ロジックが埋め込まれている、

ことを特徴とする付記６記載の記録媒体。

【０１４７】

（付記１８）

前記第１の雛型プログラムが、１つのレコードタイプをルートとし、親ノード

に関する情報を保持する各階層のレコードタイプを含むデータ構造を規定し、当該データ構造内のレコードに属性を与えるための解決ロジックを含み、

前記第２の雛型プログラムが、前記レコードに対して少なくとも追加操作、削除操作、及び検索操作を実行するための第２の雛型プログラムであって、前記所定の仕様に従った設定をレコードの属性、レコードの状態、若しくはレコードの属性及び状態の組合せで記述するための解決ロジックが埋め込まれている、

ことを特徴とする付記６記載の記録媒体。

【０１４８】

【発明の効果】

本発明により、より少ない雛型を用いて、多様なプログラムを生成できるようにするプログラム自動生成技術を提供することができた。

【図面の簡単な説明】

【図１】

プログラム自動生成装置の概要を示すブロック図である。

【図２】

データ構造解決ユニット３の概要を示すブロック図である。

【図３】

伝票型データ構造を処理するプログラムが出力する伝票操作画面の一例を示す図である。

【図４】

伝票型データ構造の模式図である。

【図５】

伝票型データ構造に対応するデータ構造解決ユニットの操作基本部に含まれる基本操作の一例を示す表である。

【図６】

伝票更新操作に対応する雛型プログラムの一例を示す図である。

【図７】

伝票型データ構造のデータ構造解決ユニットに対する解決情報を入力するための画面例を示す図である。

【図８】

データ構造を選択するための画面例を示す図である。

【図９】

伝票型データ構造のデータ構造解決ユニットに対する解決情報入力画面その１を表す図である。

【図１０】

伝票型データ構造のデータ構造解決ユニットに対する解決情報入力画面その２を表す図である。

【図１１】

伝票型データ構造のデータ構造解決ユニットに対する解決情報入力画面その３を表す図である。

【図１２】

伝票型データ構造のデータ構造解決ユニットに対する解決情報入力画面その４を表す図である。

【図１３】

解決器５の処理フローを説明するためのフローチャートである。

【図１４】

伝票型データ構造に対応する雛型プログラムの解析処理の処理フローを表すフローチャートである。

【図１５】

伝票型データ構造に対する入力画面生成及び解決情報取得処理の処理フローを表すフローチャートである。

【図１６】

伝票型データ構造を使用する他のプログラムのための解決情報入力画面の一例を示す図である。

【図１７】

時間帯予約型データ構造を使用するプログラムが出力する画面の一例を示す図である。

【図１８】

時間帯予約型データ構造の概要を示す模式図である。

【図１９】

時間帯予約型データ構造に対する基本操作の一例を示す表である。

【図２０】

時間帯予約型データ構造のデータ構造解決ユニットに対する解決情報を入力するための画面例を示す図である。

【図２１】

単純型データ構造の概要を示す模式図である。

【図２２】

マトリックス型データ構造の概要を示す模式図である。

【図２３】

自動車保険における保険商品と保障のテーブル例を示す図である。

【図２４】

図２３をマトリックス型データ構造で表した場合の一例を示す図である。

【図２５】

階層型データ構造を処理するプログラムが出力する操作画面の一例を示す図である。

【図２６】

階層型データ構造の概要を示す模式図である。

【図２７】

階層型データ構造のデータ構造解決ユニットに対する解決情報を入力するための画面例を示す図である。

【図２８】

ツリー型データ構造を処理するプログラムが出力する操作画面の一例を示す図である。

【図２９】

ツリー型データ構造の概要を示す模式図である。

【図３０】

ツリー型データ構造のデータ構造解決ユニットに対する解決情報を入力するた

めの画面例を示す図である。

【図３１】

（ａ）明細主導伝票型データ構造を説明するために用いられる売掛伝票の一例である。（ｂ）明細主導伝票型データ構造を説明するために用いられる請求伝票の一例である。（ｃ）明細主導伝票型データ構造を説明するために用いられる入金伝票の一例である。（ｄ）明細主導伝票型データ構造を説明するために用いられる取引明細の一例である。

【図３２】

明細主導伝票型データ構造を処理するプログラムが出力する操作画面の一例を示す図である。

【図３３】

明細主導伝票型データ構造の概要を示す模式図である。

【図３４】

明細主導伝票型データ構造のデータ構造解決ユニットに対する解決情報を入力するための画面例を示す図である。

【図３５】

構成型データ構造が適用可能なモデルの一例を示す図である。

【図３６】

構成型データ構造の概要を示す模式図である。

【図３７】

座席予約型データ構造が適用可能なモデルの一例を示す図である。

【図３８】

図３７のモデルに対する座席予約型データ構造を示す図である。

【図３９】

一般的な座席予約型データ構造の概要を示す模式図である。

【図４０】

系図型データ構造を適用可能なモデルの一例を説明するための図である。

【図４１】

系図型データ構造の概要を示す模式図である。

【図４２】

在庫型データ構造の概要を示す模式図である。

【図４３】

計画型データ構造が適用可能なモデルの一例を説明するための図である。

【図４４】

計画型データ構造の概要を示す模式図である。

【符号の説明】

１　プログラム自動生成装置　　３　データ構造解決ユニット

５　解決器　　７　解決ロジック解析部

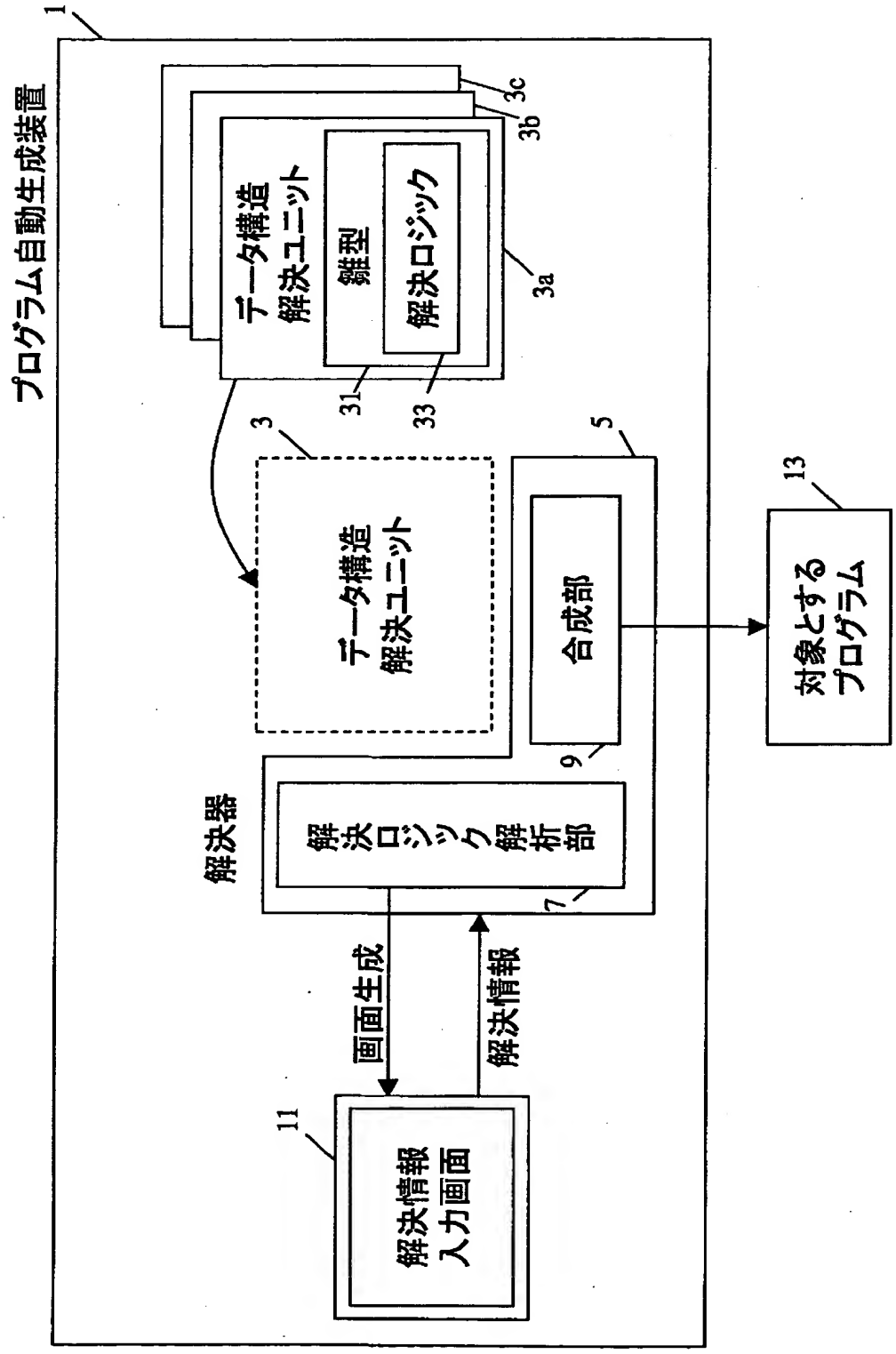
９　合成部　　１１　解決情報入力画面

１３　対象となるプログラム

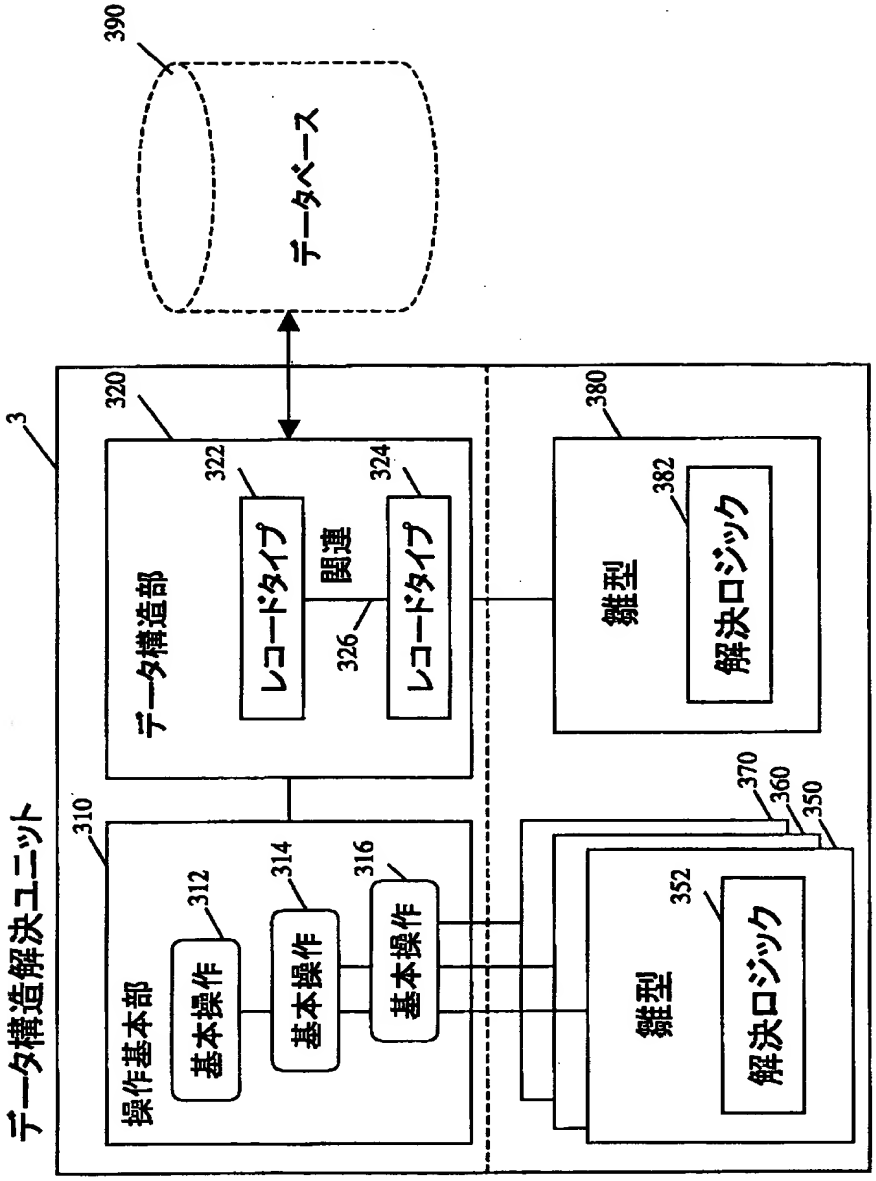
３１０　操作基本部　　３２０　データ構造部

【書類名】　　図面

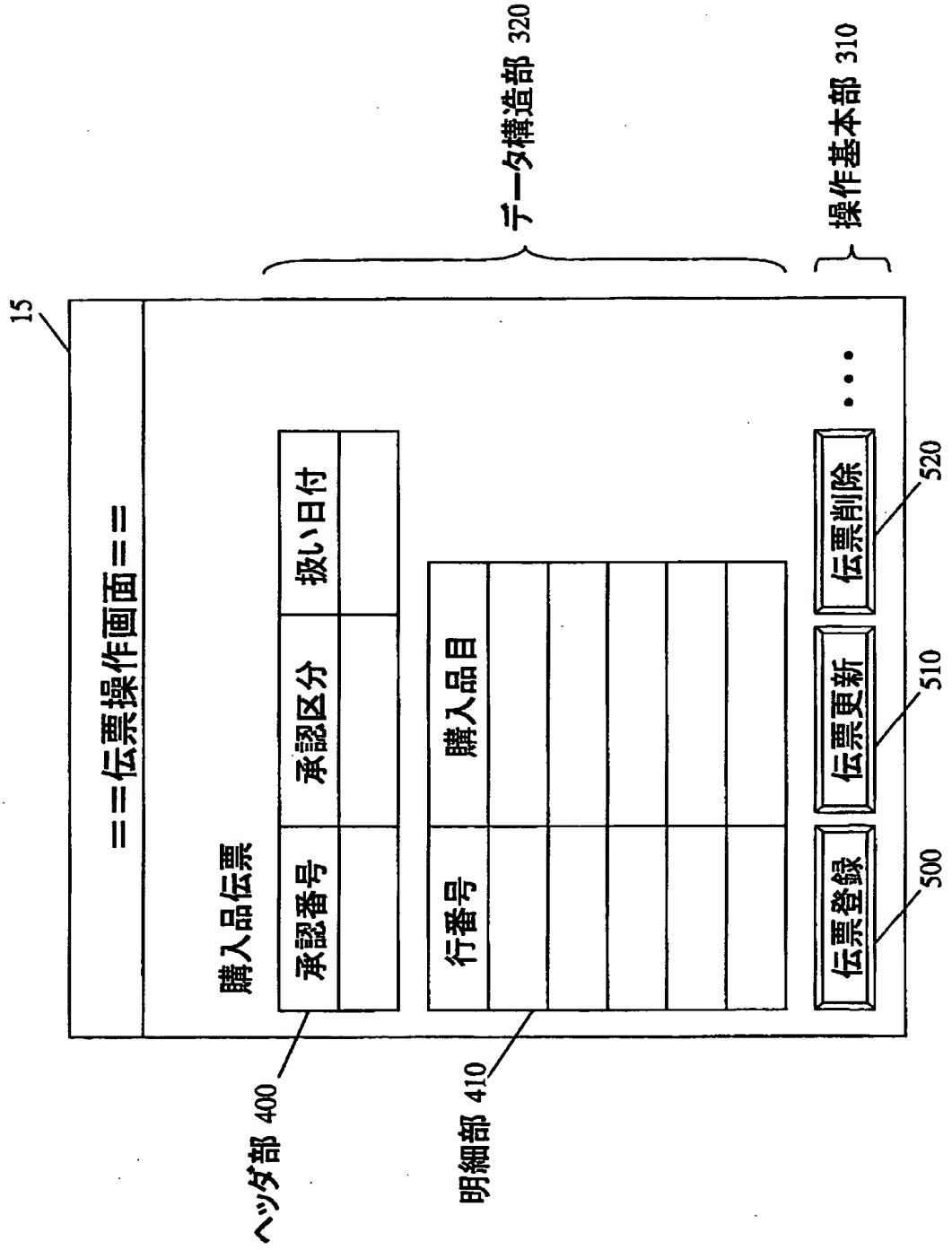
【図１】


1
プログラム自動生成装置
データ構造
解決ユニット
雛型
解決ロジック
3c
3b
3a
31
33
3
データ構造
解決ユニット
5
合成部
9
13
対象とする
プログラム
解決器
解決ロジック解析部
7
画面生成
解決情報
11
解決情報
入力画面

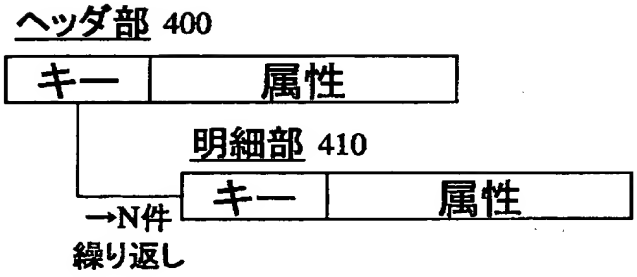
【図２】


390
データベース
3
データ構造解決ユニット
320
データ構造部
322
レコードタイプ
関連
326
324
レコードタイプ
380
雛型
382
解決ロジック
310
操作基本部
312
基本操作
314
基本操作
316
基本操作
370
360
350
雛型
352
解決ロジック

【図３】


データ構造部 320
操作基本部 310
15
＝＝伝票操作画面＝＝
購入品伝票
承認番号
承認区分
扱い日付
行番号
購入品目
伝票登録
伝票更新
伝票削除
500
510
520
ヘッダ部 400
明細部 410

【図４】



【図５】

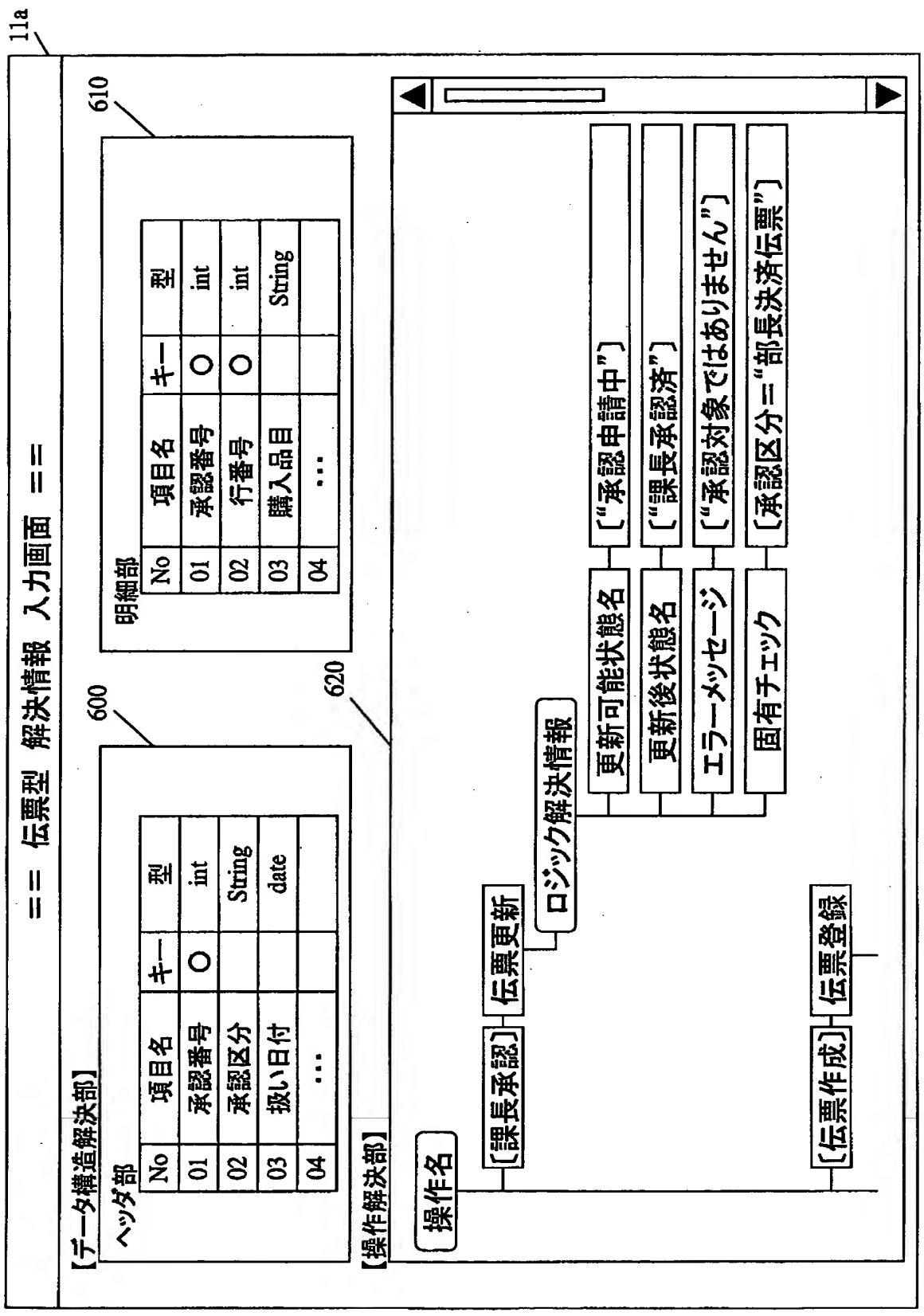
	ヘッダ部	明細部
伝票登録	登録	登録
伝票更新	更新	更新
明細追加	——	登録
・・・	・・・	・・・

【図６】

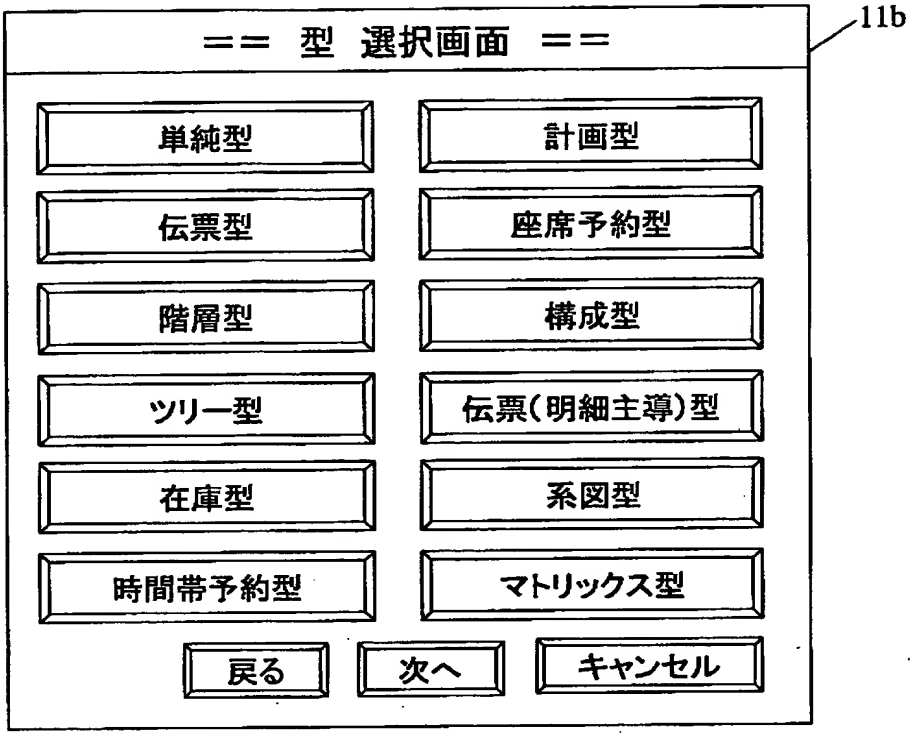
伝票更新の雛型

```
public void《操作名》(《伝票名》Msg entityInfo)
throws UserException {
  if(《更新可能な状態名》){
    mapFrom(entityInfo); //属性の更新
    setState(《更新後の状態名》);//状態の設定
    《固有チェック》
  }else{
    throw new CAAUserException(《エラーメッセージ》);
  }
}
```

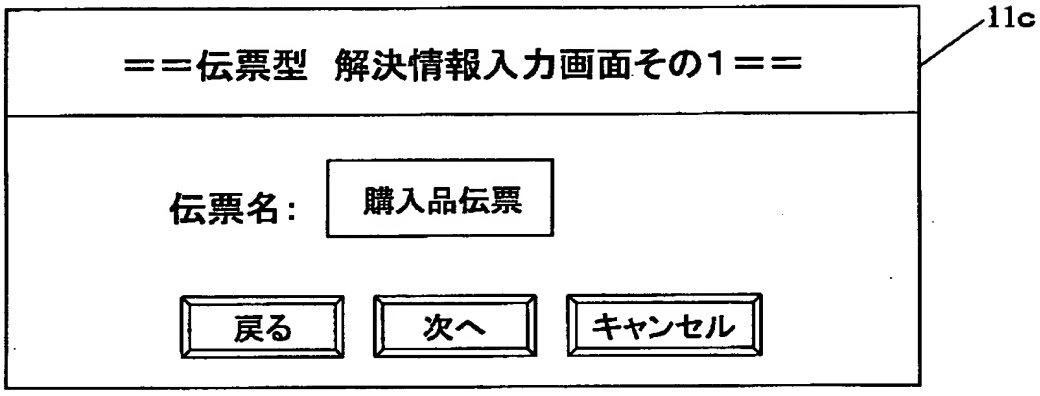
【図７】


11a
== 伝票型 解決情報 入力画面 ==
【データ構造解決部】
600
ヘッダ部
No
項目名
キー
型
01
承認番号
○
int
02
承認区分
String
03
扱い日付
date
04
…
610
明細部
No
項目名
キー
型
01
承認番号
○
int
02
行番号
○
int
03
購入品目
String
04
…
【操作解決部】
620
操作名
〔課長承認〕
伝票更新
ロジック解決情報
更新可能状態名
〔“承認申請中”〕
更新後状態名
〔“課長承認済”〕
エラーメッセージ
〔“承認対象ではありません”〕
固有チェック
〔承認区分=“部長決済伝票”〕
〔伝票作成〕
伝票登録

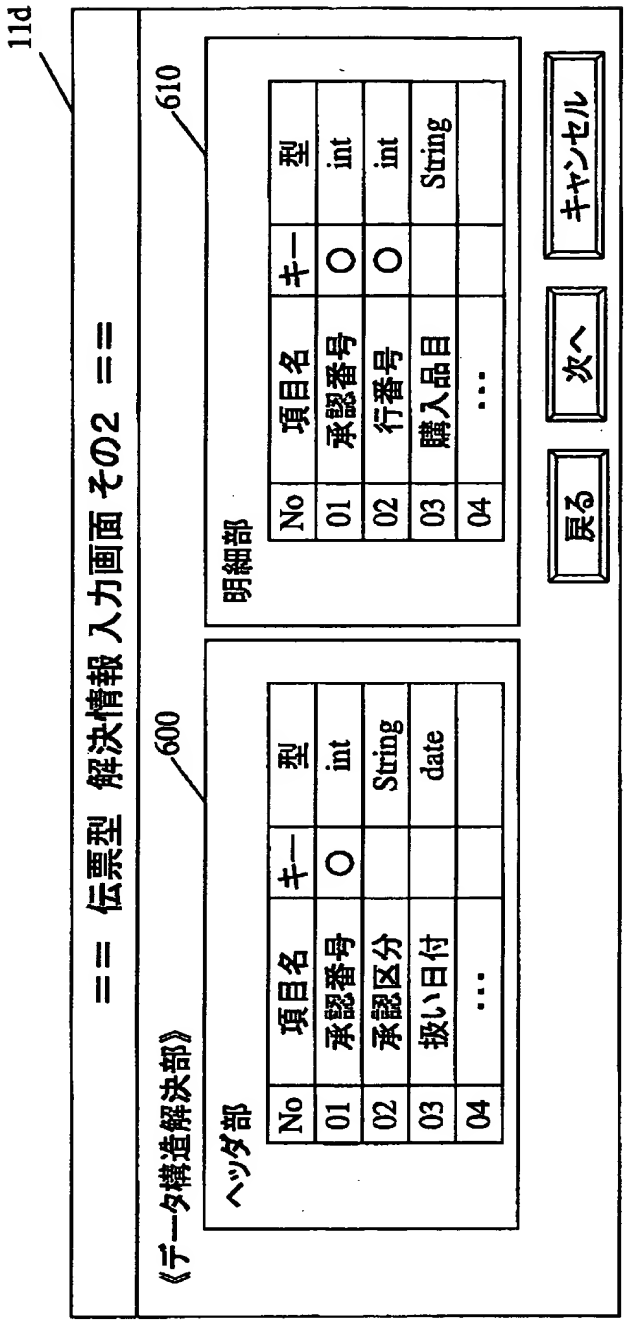
【図８】


== 型 選択画面 ==
11b
単純型
計画型
伝票型
座席予約型
階層型
構成型
ツリー型
伝票(明細主導)型
在庫型
系図型
時間帯予約型
マトリックス型
戻る
次へ
キャンセル

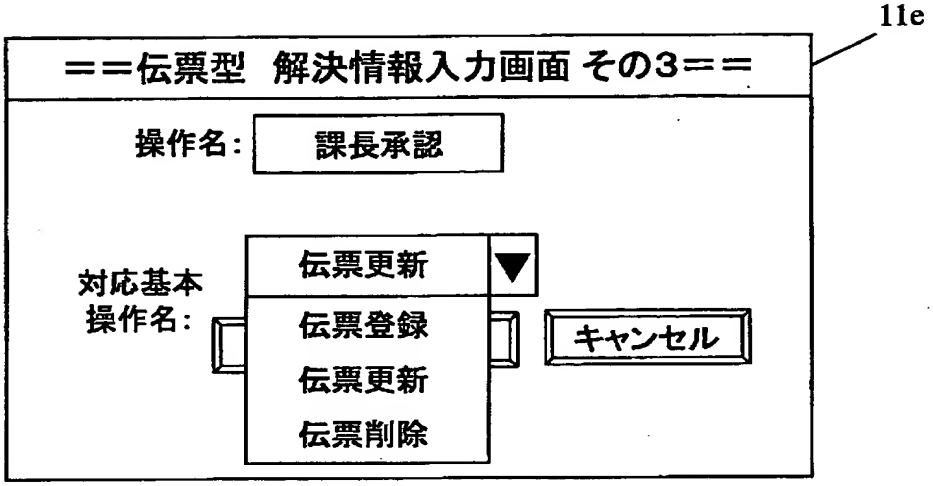
【図９】


==伝票型 解決情報入力画面その1==
11c
伝票名:
購入品伝票
戻る
次へ
キャンセル

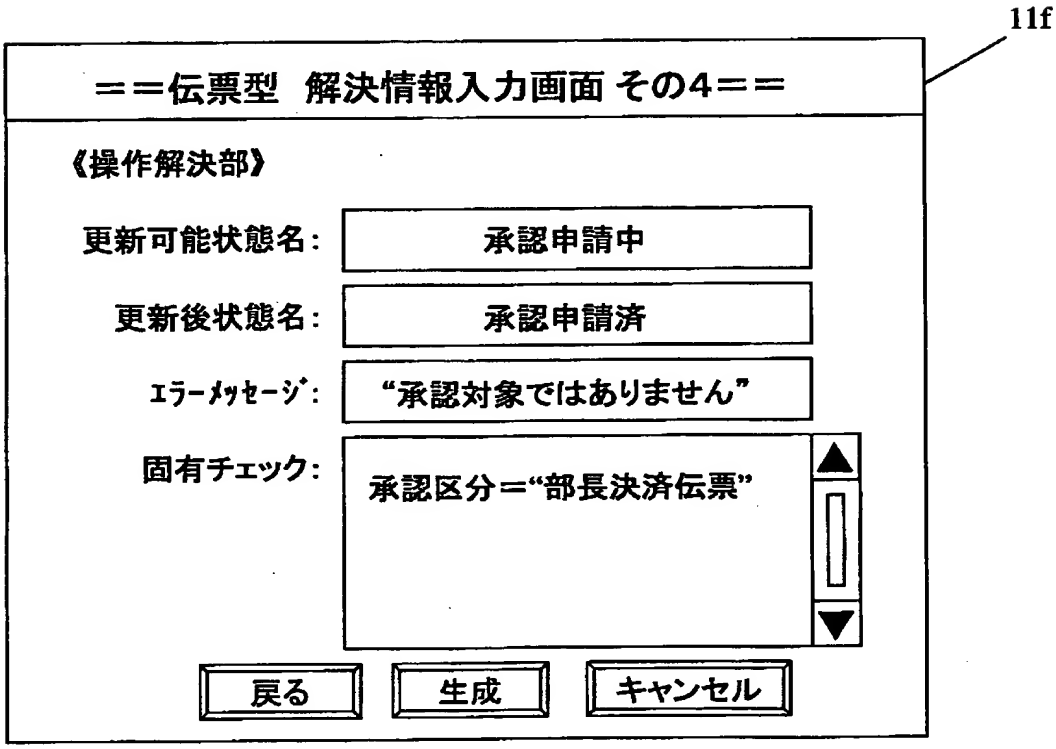
【図１０】


11d
== 伝票型 解決情報 入力画面 その2 ==
《データ構造解決部》
600
ヘッダ部
No | 項目名 | キー | 型
01 | 承認番号 | ○ | int
02 | 承認区分 | | String
03 | 扱い日付 | | date
04 | ... | |
610
明細部
No | 項目名 | キー | 型
01 | 承認番号 | ○ | int
02 | 行番号 | ○ | int
03 | 購入品目 | | String
04 | ... | |
戻る
次へ
キャンセル

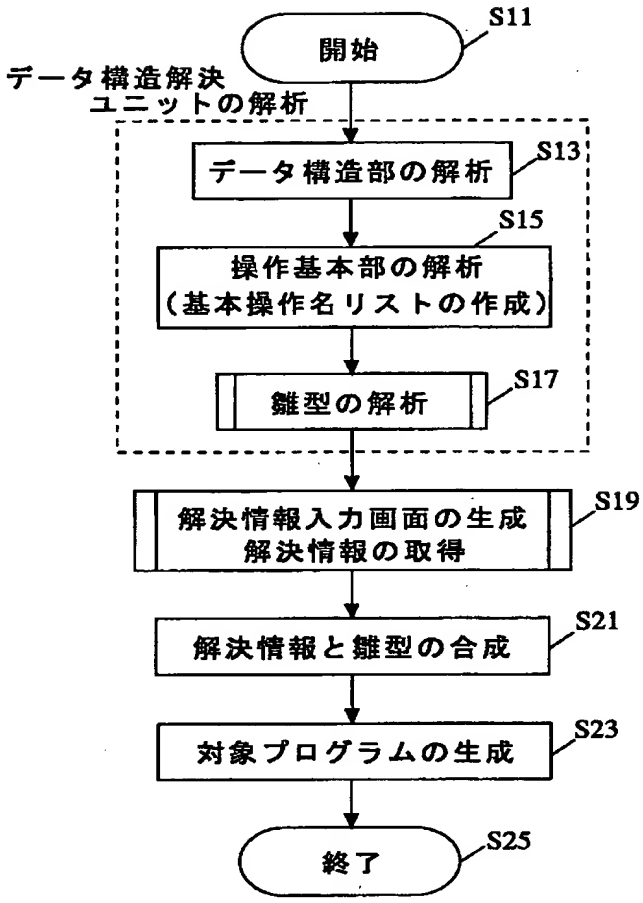
【図１１】


11e
==伝票型 解決情報入力画面 その3==
操作名:
課長承認
対応基本
操作名:
伝票更新
伝票登録
伝票更新
伝票削除
キャンセル

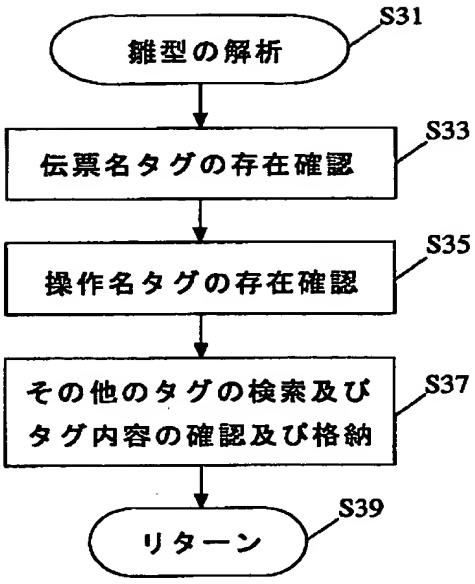
【図１２】


11f
==伝票型 解決情報入力画面 その4==
《操作解決部》
更新可能状態名:
承認申請中
更新後状態名:
承認申請済
ｴﾗｰﾒｯｾｰｼﾞ:
“承認対象ではありません”
固有チェック:
承認区分=“部長決済伝票”
戻る
生成
キャンセル

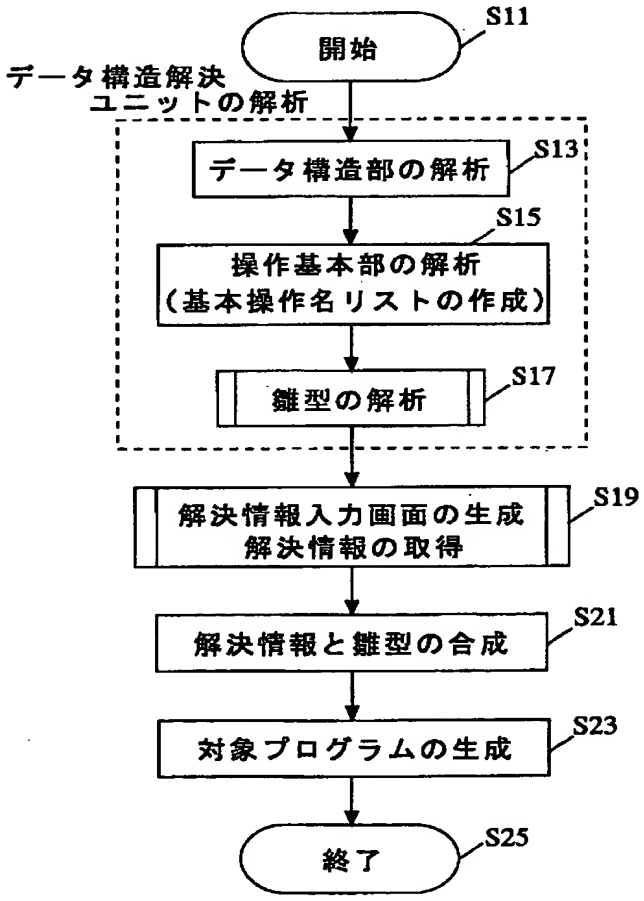
【図１３】


S11
開始
データ構造解決
ユニットの解析
S13
データ構造部の解析
S15
操作基本部の解析
（基本操作名リストの作成）
S17
雛型の解析
S19
解決情報入力画面の生成
解決情報の取得
S21
解決情報と雛型の合成
S23
対象プログラムの生成
S25
終了

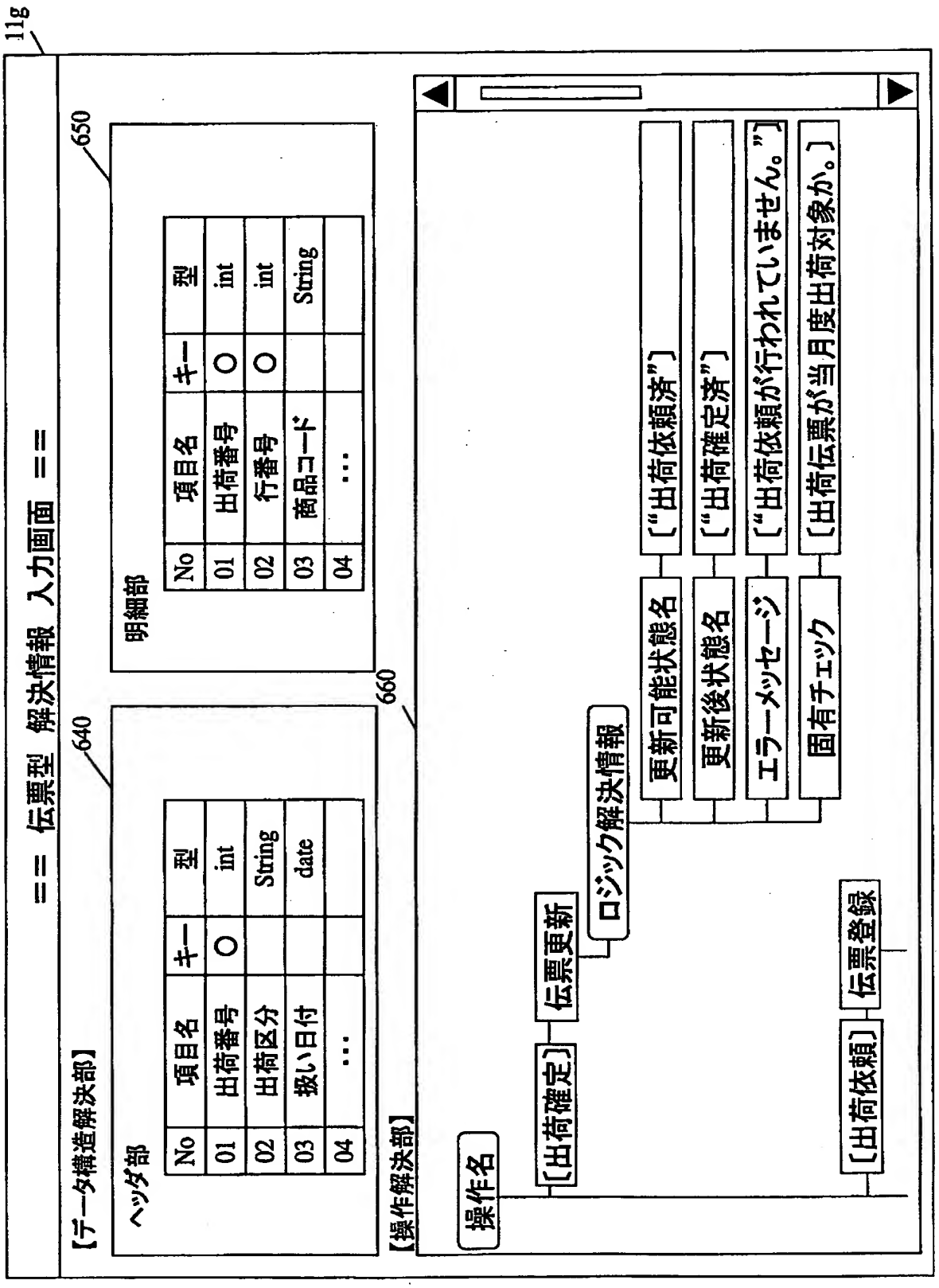
【図１４】


S31
雛型の解析
S33
伝票名タグの存在確認
S35
操作名タグの存在確認
S37
その他のタグの検索及び
タグ内容の確認及び格納
S39
リターン

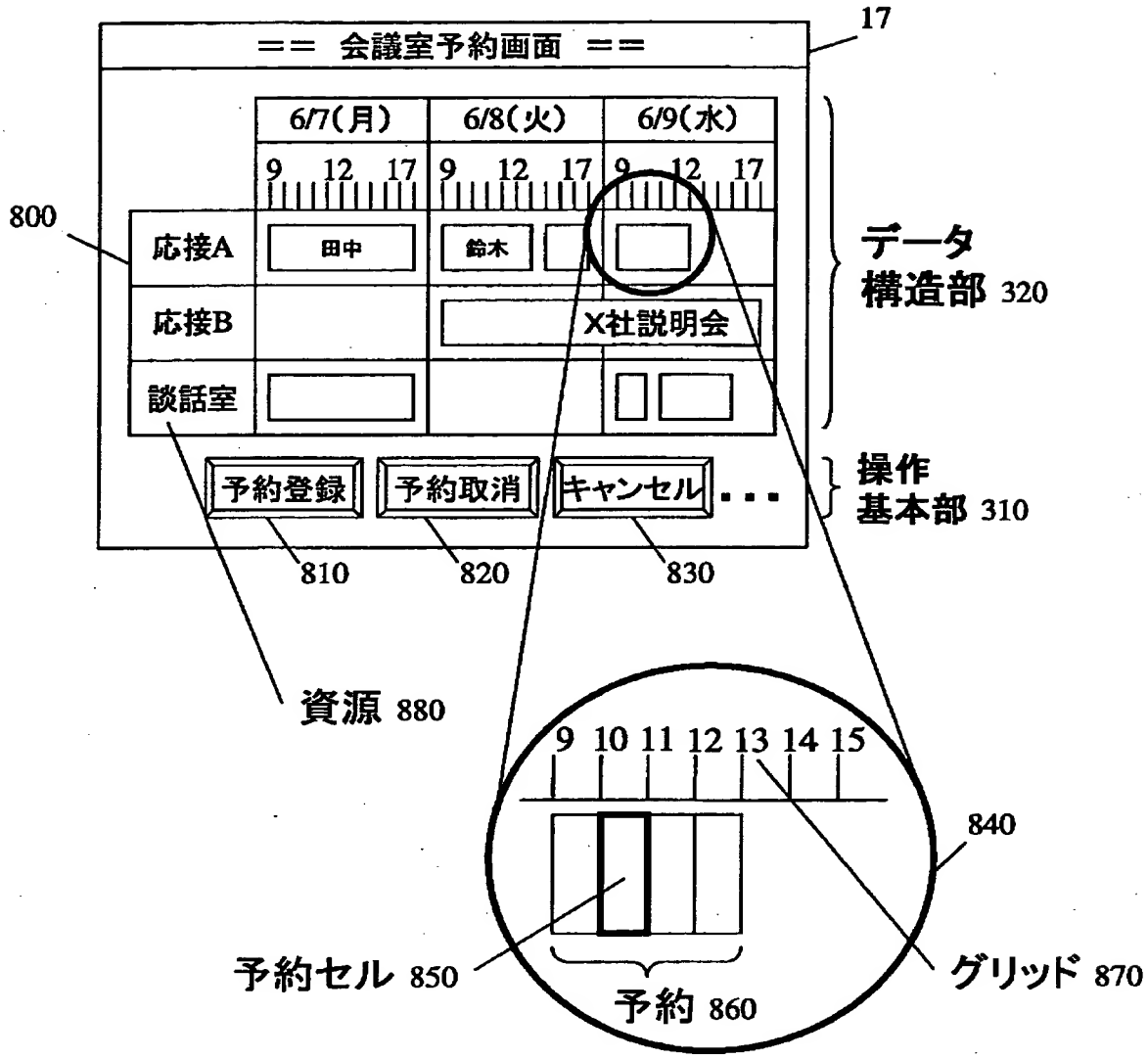
【図１５】


S11
開始
データ構造解決
ユニットの解析
S13
データ構造部の解析
S15
操作基本部の解析
(基本操作名リストの作成)
S17
雛型の解析
S19
解決情報入力画面の生成
解決情報の取得
S21
解決情報と雛型の合成
S23
対象プログラムの生成
S25
終了

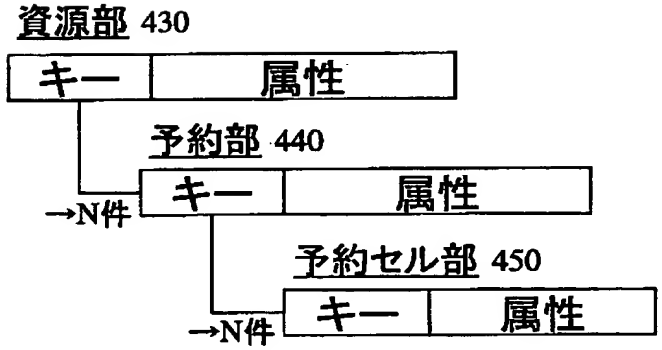
【図１６】


11g
== 伝票型 解決情報 入力画面 ==
【データ構造解決部】
640
ヘッダ部
No | 項目名 | キー | 型
01 | 出荷番号 | ○ | int
02 | 出荷区分 | | String
03 | 扱い日付 | | date
04 | … | | |
650
明細部
No | 項目名 | キー | 型
01 | 出荷番号 | ○ | int
02 | 行番号 | ○ | int
03 | 商品コード | | String
04 | … | | |
660
【操作解決部】
操作名
〔出荷確定〕
伝票更新
ロジック解決情報
更新可能状態名
〔“出荷依頼済”〕
更新後状態名
〔“出荷確定済”〕
エラーメッセージ
〔“出荷依頼が行われていません。”〕
固有チェック
〔出荷伝票が当月度出荷対象か。〕
〔出荷依頼〕
伝票登録

【図１７】


== 会議室予約画面 ==
17
6/7(月)
6/8(火)
6/9(水)
9 12 17
9 12 17
9 12 17
800
応接A
田中
鈴木
応接B
X社説明会
談話室
データ
構造部 320
予約登録
予約取消
キャンセル
操作
基本部 310
810
820
830
資源 880
9 10 11 12 13 14 15
840
予約セル 850
予約 860
グリッド 870

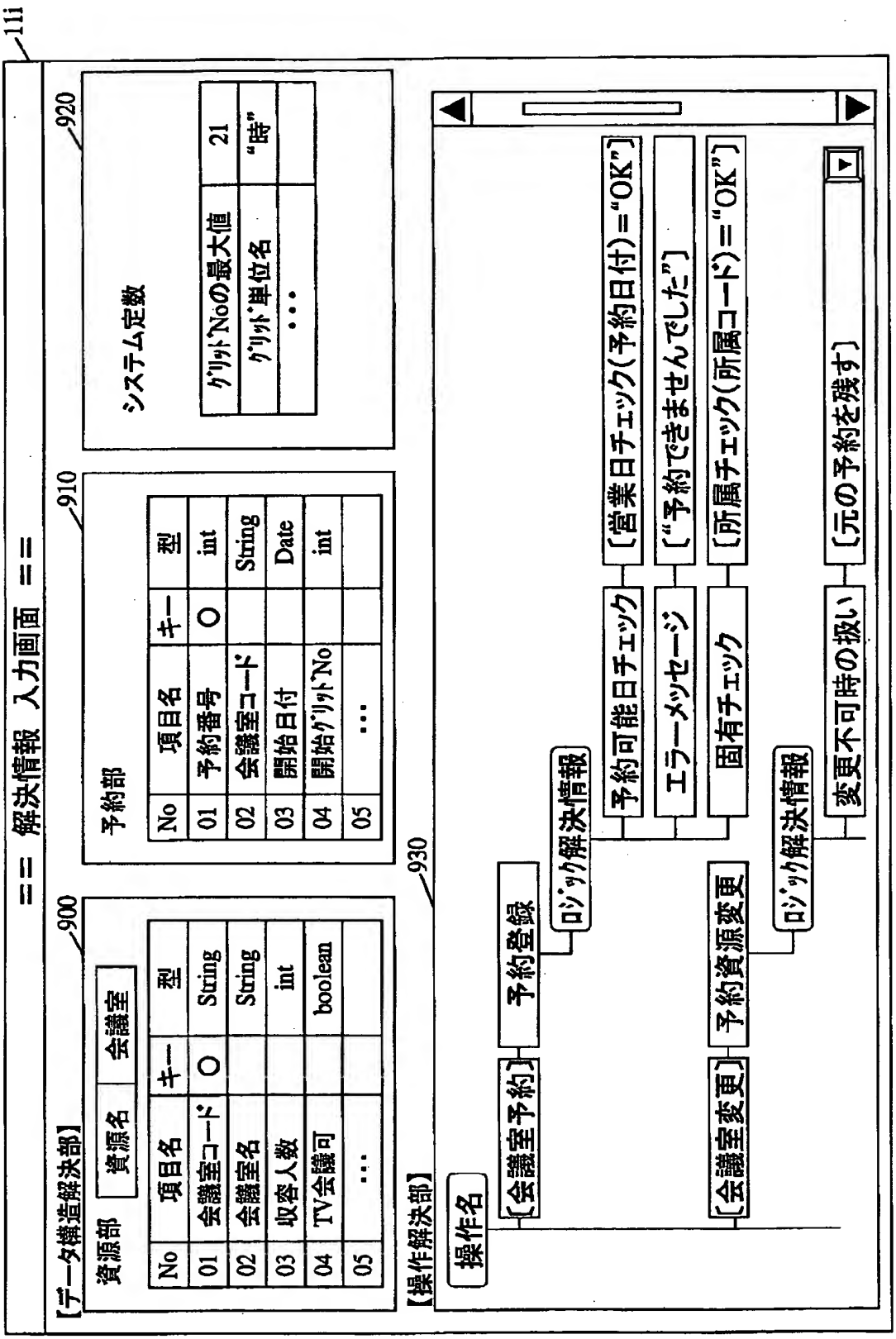
【図１８】


資源部 430
キー
属性
予約部 440
→N件
キー
属性
予約セル部 450
→N件
キー
属性

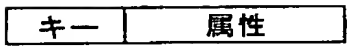
【図１９】

	資源部	予約部	予約セル部
予約登録	登録	登録	登録
予約取消	削除	削除	削除
空き状況確認	―	検索	―
予約期間変更	―	更新	削除/登録
予約資源変更	検索	更新	削除/登録
予約条件検索	―	検索	―
・・・	・・・	・・・	・・・

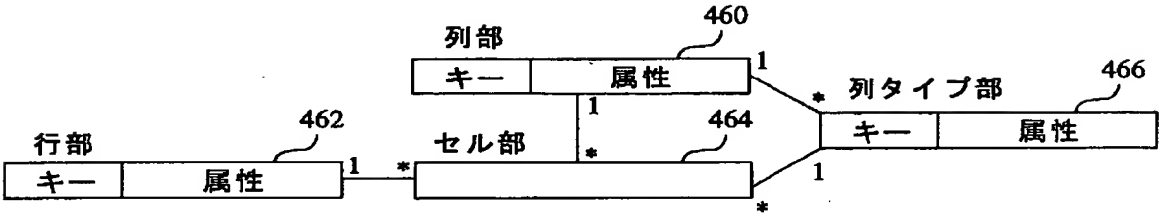
【図２０】


111
ニニ 解決情報 入力画面 ニニ
【データ構造解決部】
900
資源部
資源名
会議室
No
項目名
キー
型
01
会議室コード
O
String
02
会議室名
String
03
収容人数
int
04
TV会議可
boolean
05
…
910
予約部
No
項目名
キー
型
01
予約番号
O
int
02
会議室コード
String
03
開始日付
Date
04
開始ｸﾘｯﾄNo
int
05
…
920
システム定数
ｸﾘｯﾄNoの最大値
21
ｸﾘｯﾄ単位名
“時”
…
930
【操作解決部】
操作名
〔会議室予約〕
予約登録
ﾛｼﾞｯｸ解決情報
予約可能日チェック
〔営業日チェック(予約日付)＝“OK”〕
エラーメッセージ
〔“予約できませんでした”〕
〔会議室変更〕
予約資源変更
ﾛｼﾞｯｸ解決情報
固有チェック
〔所属チェック(所属コード)＝“OK”〕
変更不可時の扱い
〔元の予約を残す〕

【図２１】



【図２２】

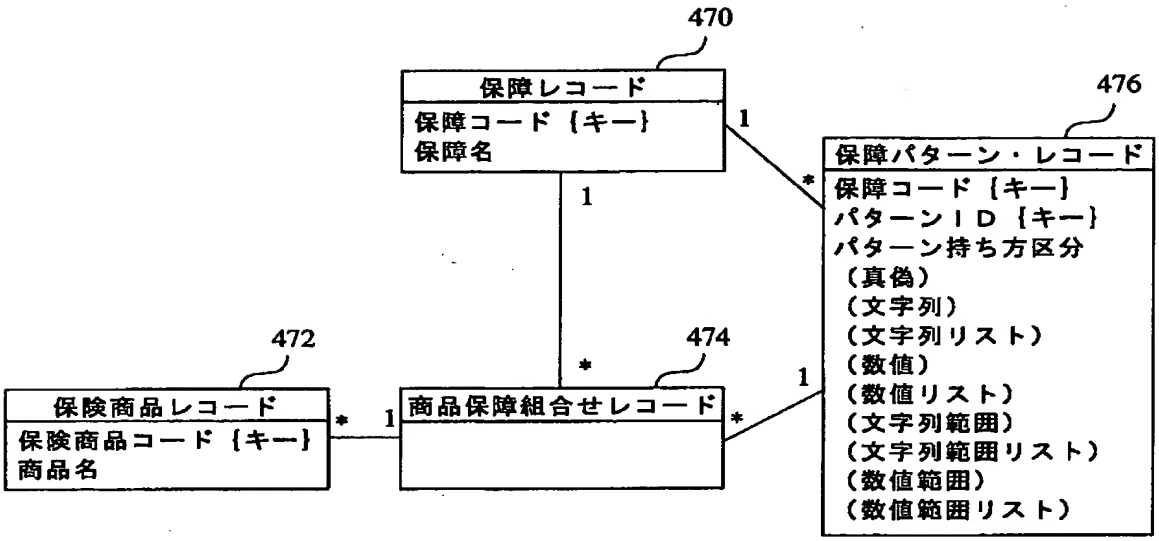


【図２３】

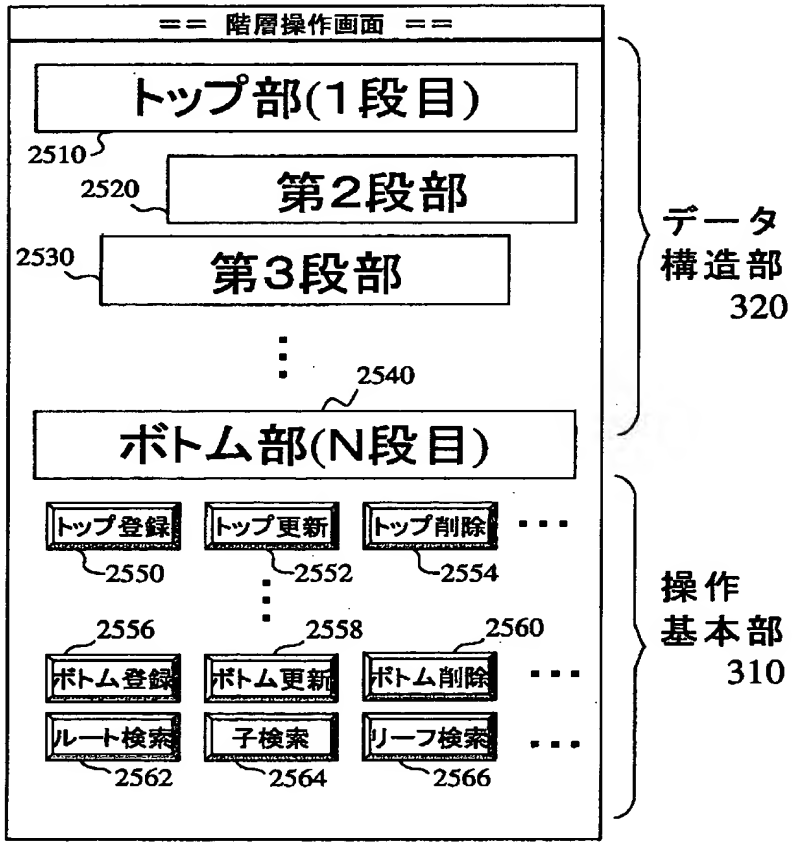
保障／保険商品	用途・車種	年齢別不担保特約	ファミリーバイク特約	無保険車傷害保険
A	・自家用普通乗用車 ・自家用小型乗用車 ・自家用軽四輪乗用車 ・二輪自動車 ・原付自動車	・21歳 ・26歳 ・30歳 ・年齢を問わず	あり	あり
B	・自家用普通乗用車 ・自家用小型乗用車 ・自家用軽四輪乗用車	・21歳 ・26歳 ・30歳 ・年齢を問わず	あり	あり
C	・自家用普通乗用車 ・自家用小型乗用車 ・自家用軽四輪乗用車	・21歳 ・26歳 ・30歳 ・年齢を問わず	あり	あり
D	・自家用普通乗用車 ・自家用小型乗用車 ・自家用軽四輪乗用車	・21歳 ・26歳 ・30歳 ・年齢を問わず	なし	なし
E	・自家用普通乗用車 ・自家用小型乗用車 ・自家用軽四輪乗用車 ・二輪自動車 ・原付自動車	・50歳	なし	なし

出証特２０００－３１０４７２１

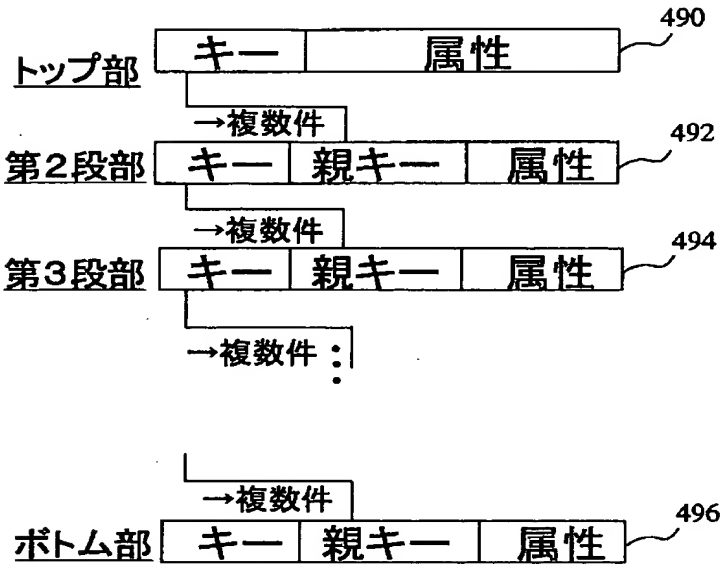
【図２４】


470
保障レコード
保障コード｛キー｝
保障名
476
保障パターン・レコード
保障コード｛キー｝
パターンＩＤ｛キー｝
パターン持ち方区分
（真偽）
（文字列）
（文字列リスト）
（数値）
（数値リスト）
（文字列範囲）
（文字列範囲リスト）
（数値範囲）
（数値範囲リスト）
472
保険商品レコード
保険商品コード｛キー｝
商品名
474
商品保障組合せレコード
1
*

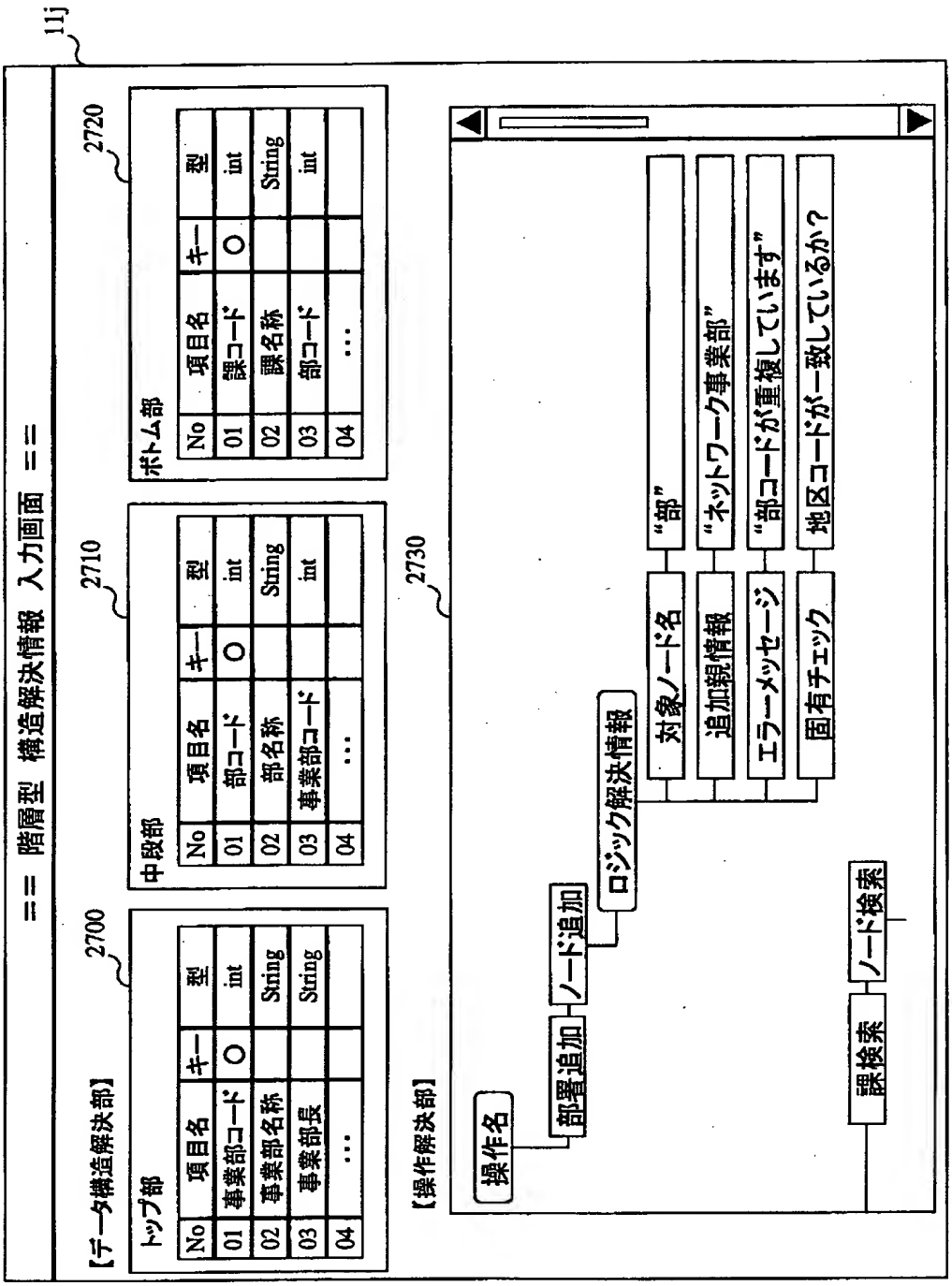
【図２５】


== 階層操作画面 ==
トップ部(1段目)
2510
第2段部
2520
第3段部
2530
2540
ボトム部(N段目)
データ構造部 320
トップ登録
トップ更新
トップ削除
2550
2552
2554
2556
2558
2560
ボトム登録
ボトム更新
ボトム削除
ルート検索
子検索
リーフ検索
2562
2564
2566
操作基本部 310

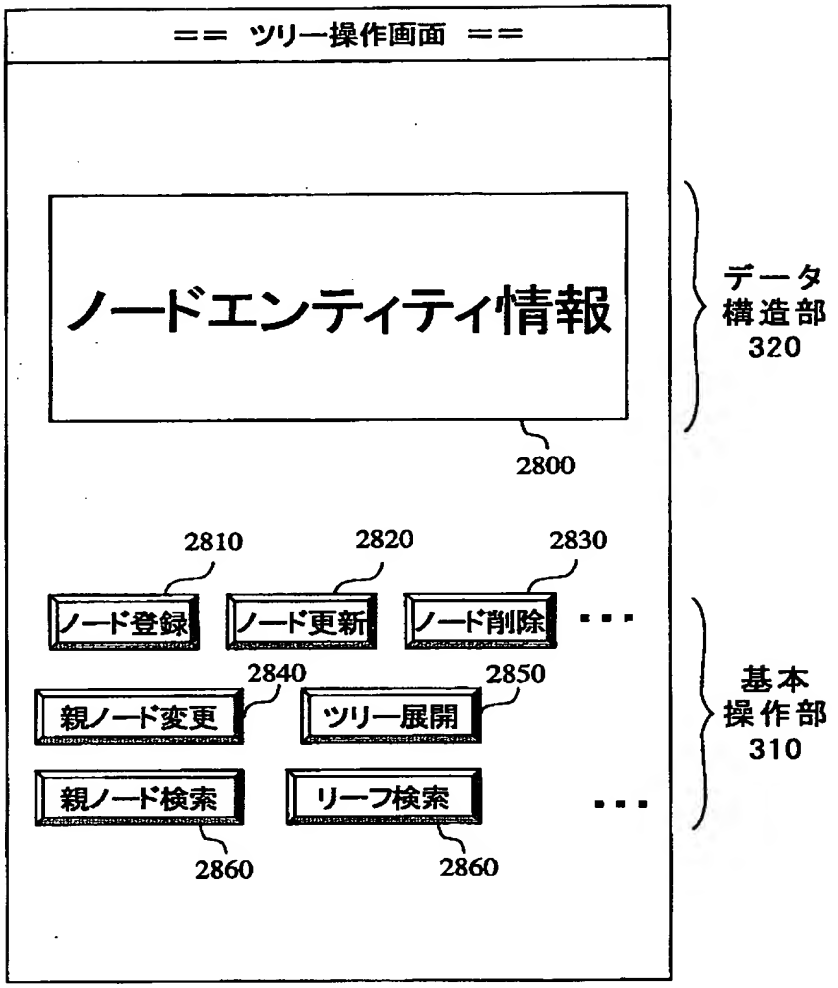
【図２６】


490
トップ部
キー
属性
→複数件
492
第2段部
キー
親キー
属性
→複数件
494
第3段部
キー
親キー
属性
→複数件
→複数件
496
ボトム部
キー
親キー
属性

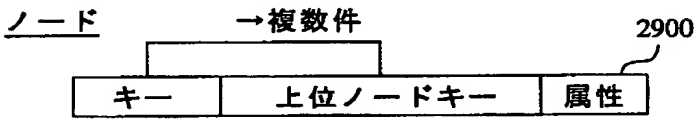
【図２７】


11j
== 階層型 構造解決情報 入力画面 ==
【データ構造解決部】
2700
トップ部
No | 項目名 | キー | 型
01 | 事業部コード | ○ | int
02 | 事業部名称 | | String
03 | 事業部長 | | String
04 | … | |
2710
中段部
No | 項目名 | キー | 型
01 | 部コード | ○ | int
02 | 部名称 | | String
03 | 事業部コード | | int
04 | … | |
2720
ボトム部
No | 項目名 | キー | 型
01 | 課コード | ○ | int
02 | 課名称 | | String
03 | 部コード | | int
04 | … | |
【操作解決部】
2730
操作名
部署追加
ノード追加
ロジック解決情報
対象ノード名
追加親情報
エラーメッセージ
固有チェック
“部”
“ネットワーク事業部”
“部コードが重複しています”
地区コードが一致しているか？
課検索
ノード検索

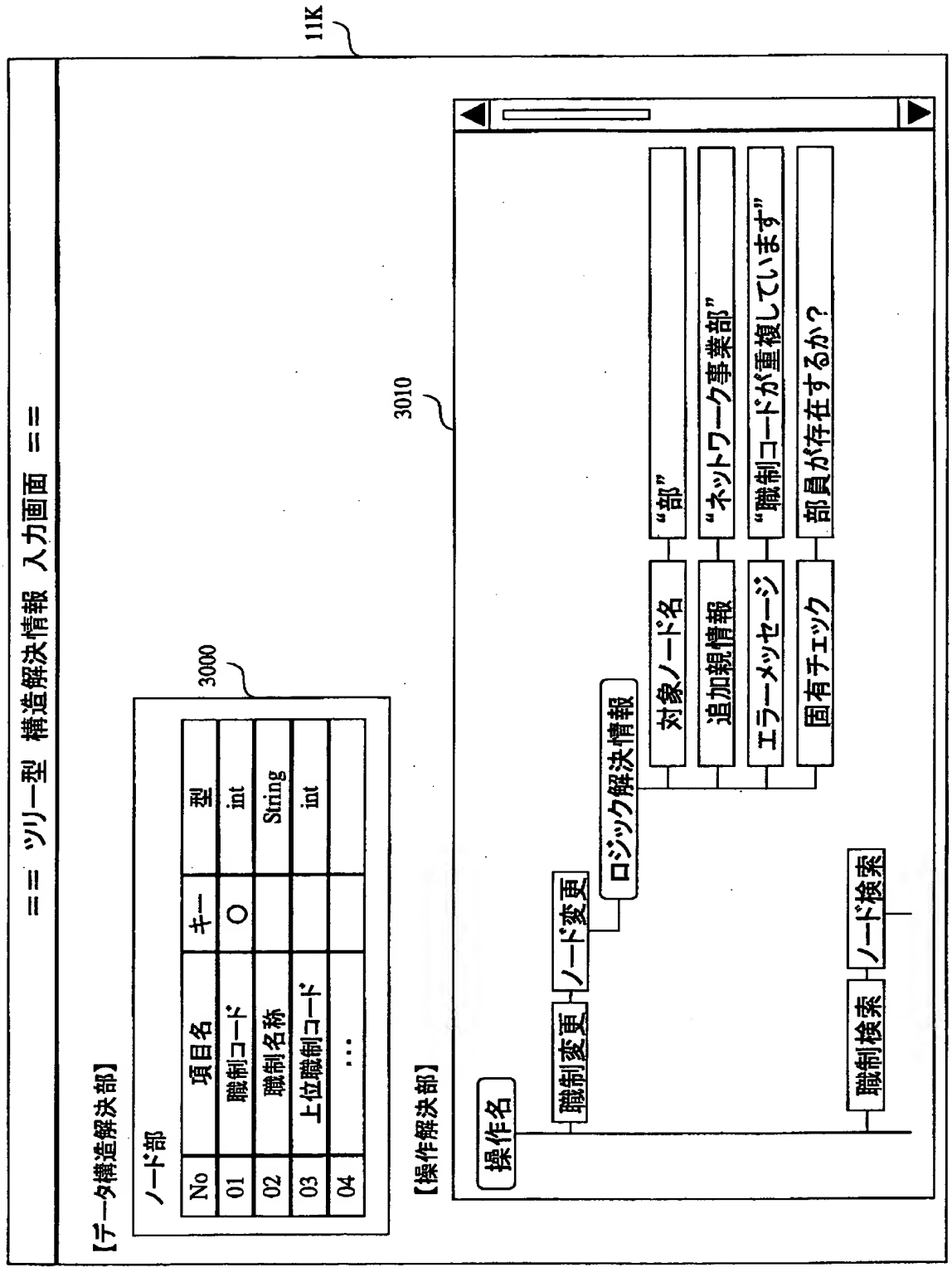
【図２８】


== ツリー操作画面 ==
ノードエンティティ情報
2800
データ構造部 320
2810
2820
2830
ノード登録
ノード更新
ノード削除
2840
2850
親ノード変更
ツリー展開
親ノード検索
リーフ検索
2860
2860
基本操作部 310

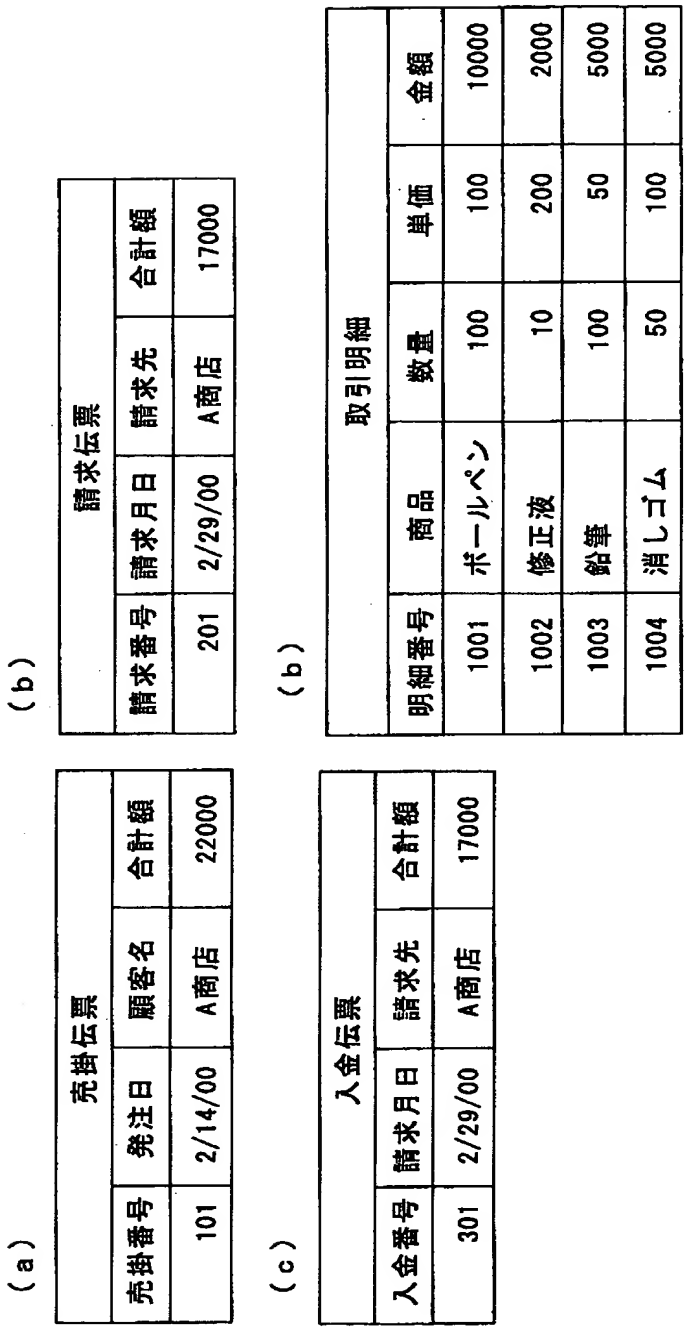
【図２９】


ノード
→複数件
2900
キー
上位ノードキー
属性

【図３０】


11K
＝＝ ツリー型 構造解決情報 入力画面 ＝＝
【データ構造解決部】
3000
ノード部
No
項目名
キー
型
01
職制コード
○
int
02
職制名称
String
03
上位職制コード
int
04
…
3010
【操作解決部】
操作名
職制変更
ノード変更
ロジック解決情報
対象ノード名
追加親情報
エラーメッセージ
固有チェック
“部”
“ネットワーク事業部”
“職制コードが重複しています”
部員が存在するか？
職制検索
ノード検索

【図３１】



（ａ）

売掛伝票			
売掛番号	発注日	顧客名	合計額
101	2/14/00	A商店	22000

（ｂ）

請求伝票			
請求番号	請求月日	請求先	合計額
201	2/29/00	A商店	17000

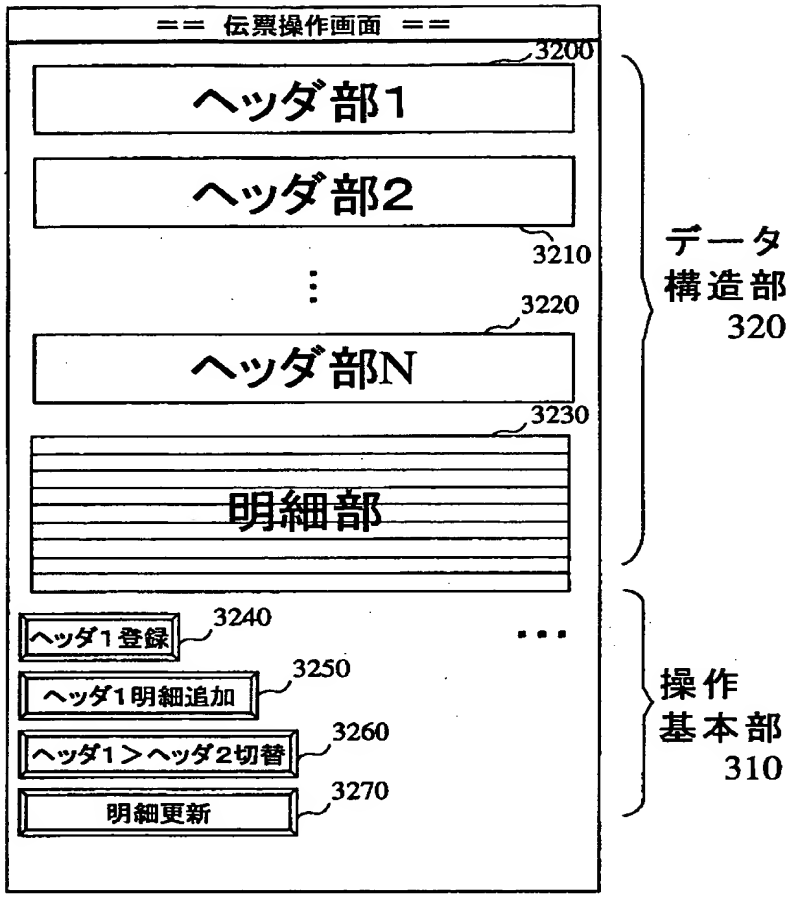
（ｃ）

入金伝票			
入金番号	請求月日	請求先	合計額
301	2/29/00	A商店	17000

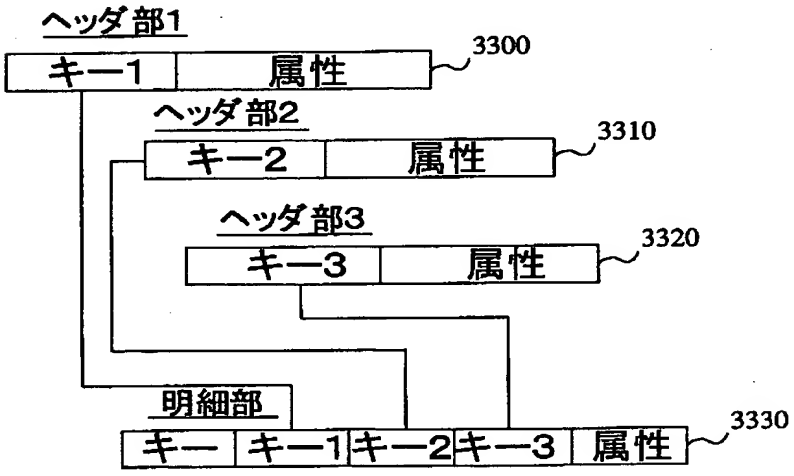
（ｂ）

取引明細				
明細番号	商品	数量	単価	金額
1001	ボールペン	100	100	10000
1002	修正液	10	200	2000
1003	鉛筆	100	50	5000
1004	消しゴム	50	100	5000

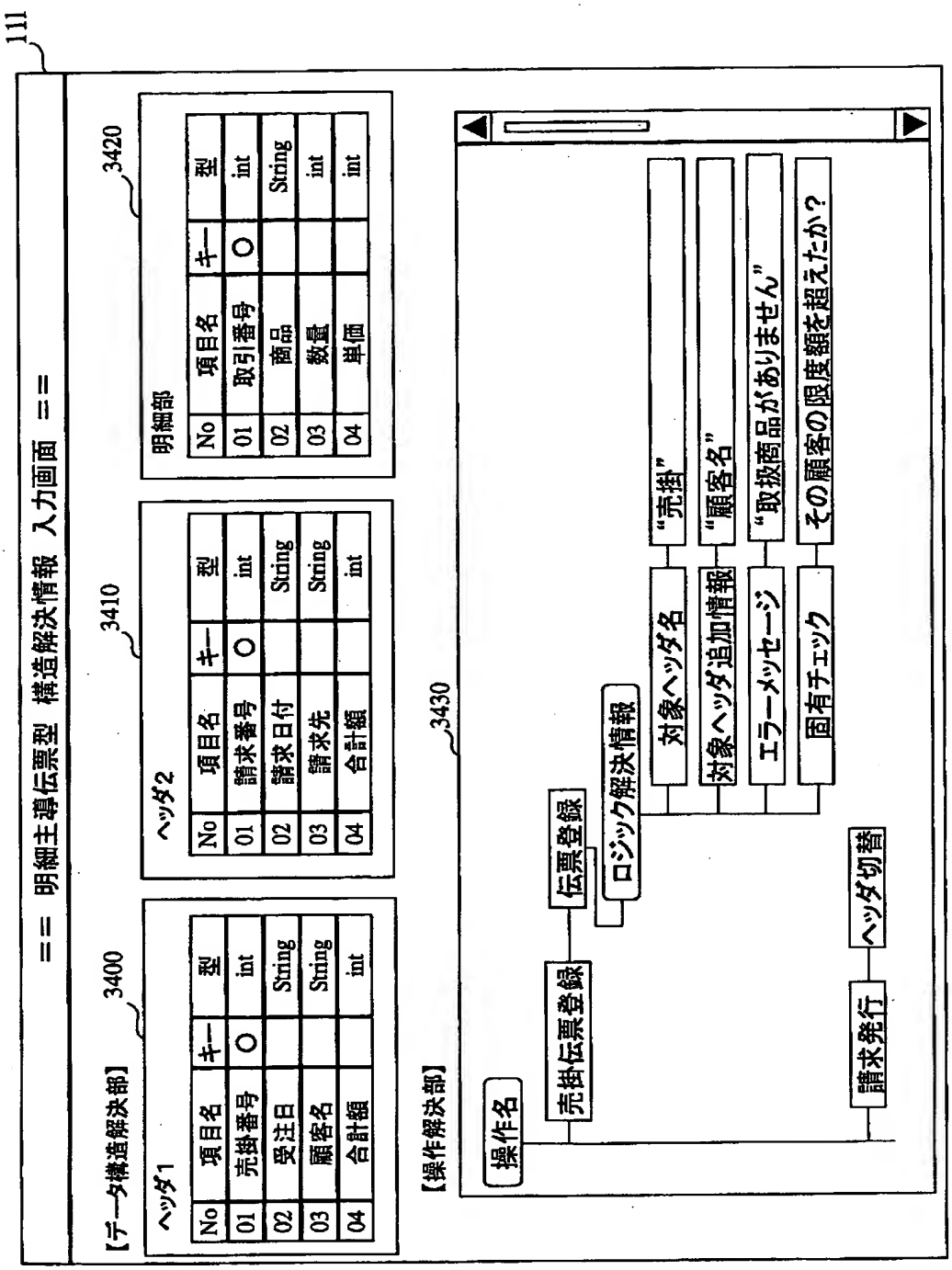
【図３２】


== 伝票操作画面 ==
3200
ヘッダ部1
ヘッダ部2
3210
3220
ヘッダ部N
3230
明細部
データ構造部 320
3240
ヘッダ1登録
3250
ヘッダ1明細追加
3260
ヘッダ1＞ヘッダ2切替
3270
明細更新
操作基本部 310

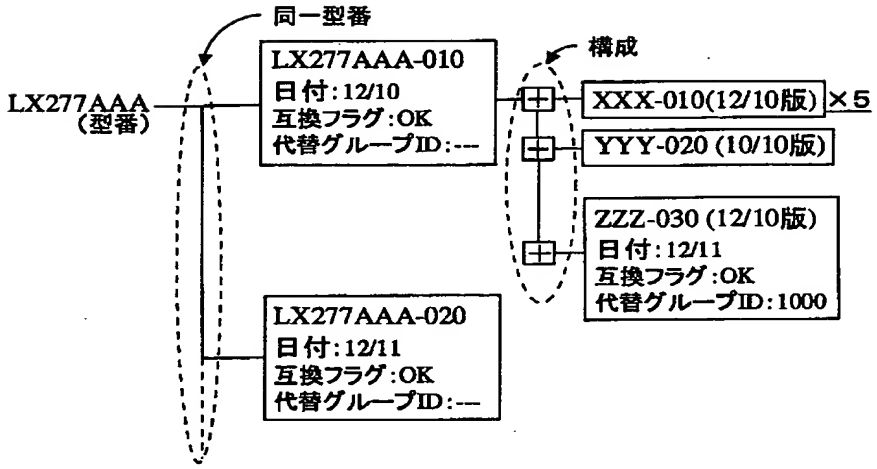
【図３３】


ヘッダ部1
キー1
属性
3300
ヘッダ部2
キー2
属性
3310
ヘッダ部3
キー3
属性
3320
明細部
キー
キー1
キー2
キー3
属性
3330

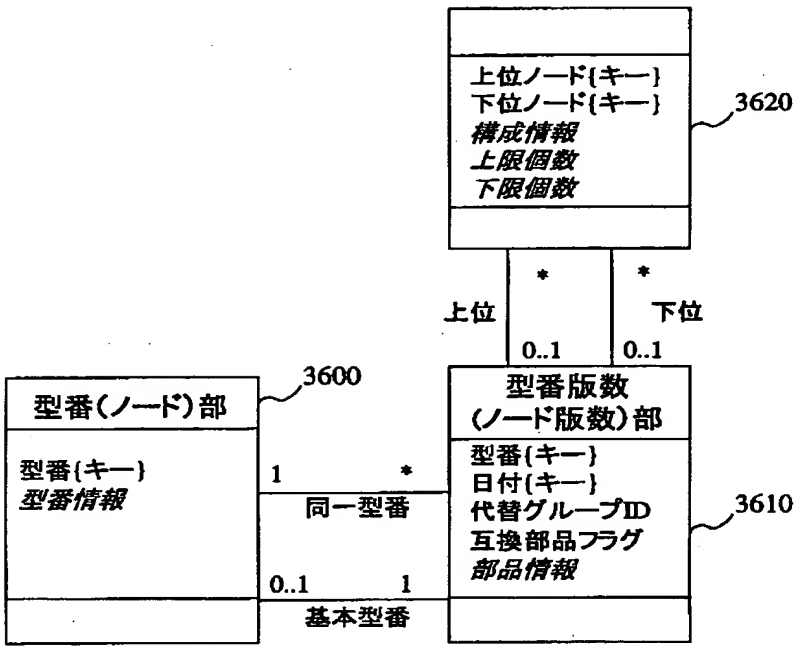
【図３４】


111
== 明細主導伝票型 構造解決情報 入力画面 ==
【データ構造解決部】
3400
ヘッダ1
No 項目名 キー 型
01 売掛番号 ○ int
02 受注日 String
03 顧客名 String
04 合計額 int
3410
ヘッダ2
No 項目名 キー 型
01 請求番号 ○ int
02 請求日付 String
03 請求先 String
04 合計額 int
3420
明細部
No 項目名 キー 型
01 取引番号 ○ int
02 商品 String
03 数量 int
04 単価 int
【操作解決部】
3430
操作名
売掛伝票登録
伝票登録
ロジック解決情報
対象ヘッダ名
対象ヘッダ追加情報
エラーメッセージ
固有チェック
ヘッダ切替
請求発行
"売掛"
"顧客名"
"取扱商品がありません"
その顧客の限度額を超えたか？

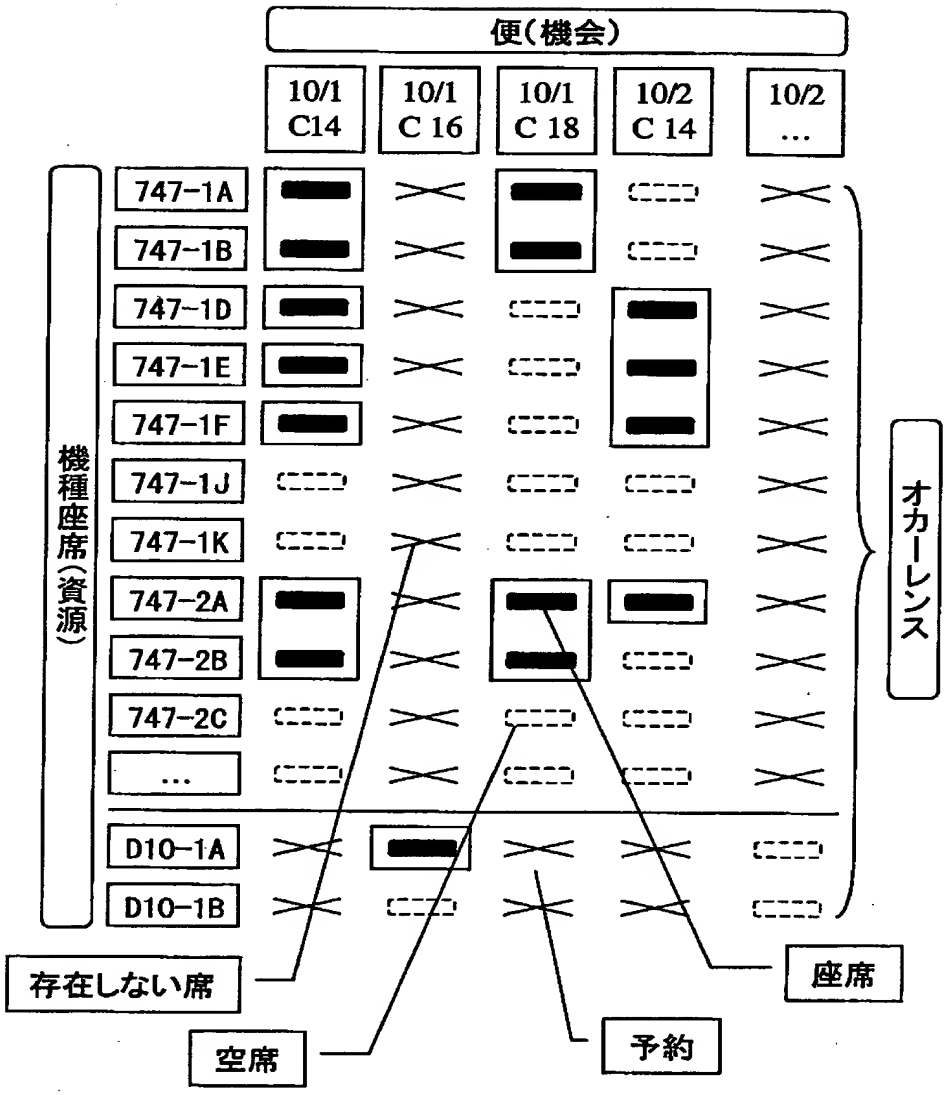
【図３５】


同一型番
構成
LX277AAA
(型番)
LX277AAA-010
日付:12/10
互換フラグ:OK
代替グループID:---
XXX-010(12/10版)
×5
YYY-020 (10/10版)
ZZZ-030 (12/10版)
日付:12/11
互換フラグ:OK
代替グループID:1000
LX277AAA-020
日付:12/11
互換フラグ:OK
代替グループID:---

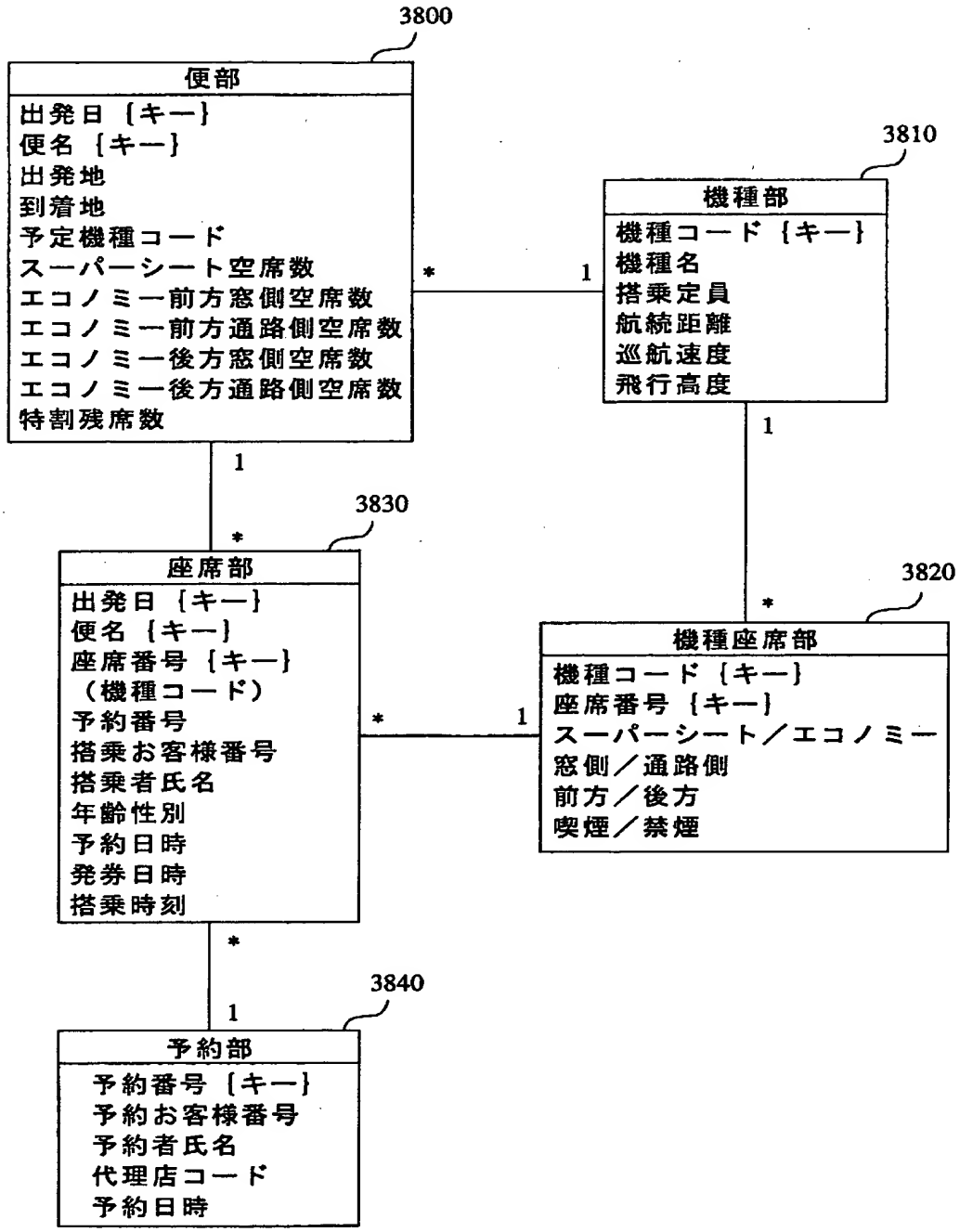
【図３６】


上位ノード{キー}
下位ノード{キー}
構成情報
上限個数
下限個数
3620
*
*
上位
下位
0..1
0..1
型番(ノード)部
3600
型番{キー}
型番情報
1
*
同一型番
0..1
1
基本型番
型番版数
(ノード版数)部
型番{キー}
日付{キー}
代替グループID
互換部品フラグ
部品情報
3610

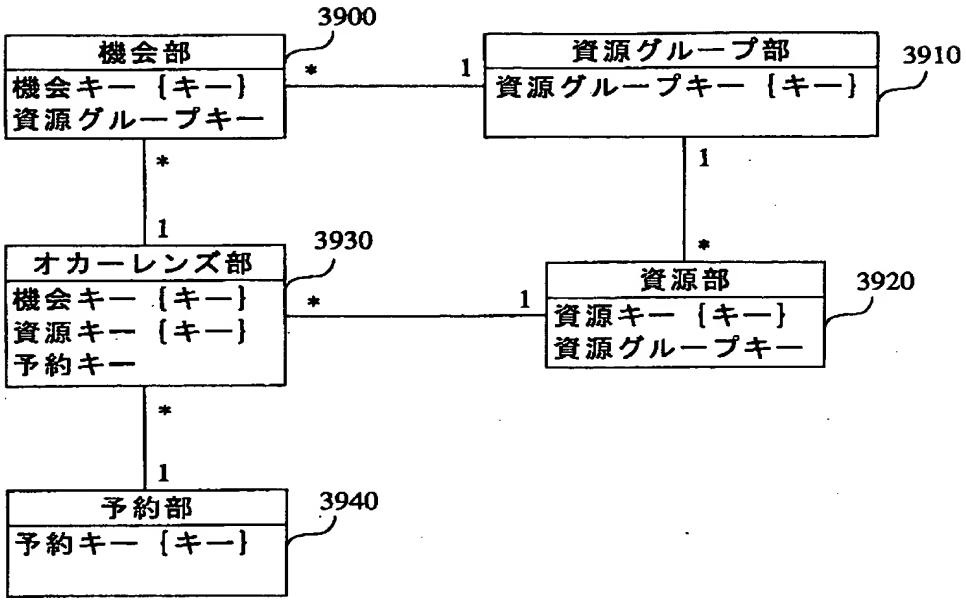
【図３７】


便(機会)
10/1 C14
10/1 C 16
10/1 C 18
10/2 C 14
10/2 ...
機種座席(資源)
747-1A
747-1B
747-1D
747-1E
747-1F
747-1J
747-1K
747-2A
747-2B
747-2C
...
D10-1A
D10-1B
オカーレンス
存在しない席
空席
予約
座席

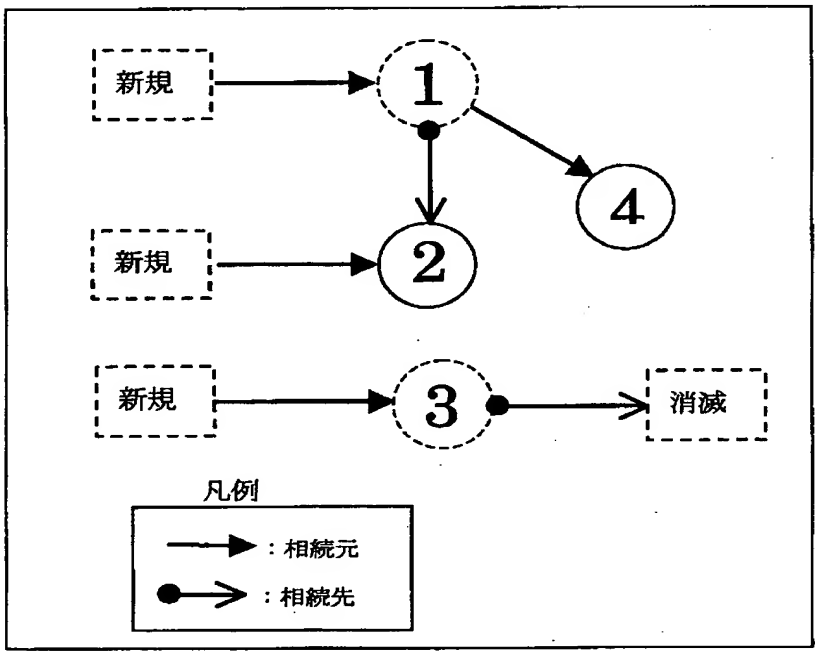
【図３８】


3800
便部
出発日〔キー〕
便名〔キー〕
出発地
到着地
予定機種コード
スーパーシート空席数
エコノミー前方窓側空席数
エコノミー前方通路側空席数
エコノミー後方窓側空席数
エコノミー後方通路側空席数
特割残席数
3810
機種部
機種コード〔キー〕
機種名
搭乗定員
航続距離
巡航速度
飛行高度
3830
座席部
出発日〔キー〕
便名〔キー〕
座席番号〔キー〕
（機種コード）
予約番号
搭乗お客様番号
搭乗者氏名
年齢性別
予約日時
発券日時
搭乗時刻
3820
機種座席部
機種コード〔キー〕
座席番号〔キー〕
スーパーシート／エコノミー
窓側／通路側
前方／後方
喫煙／禁煙
3840
予約部
予約番号〔キー〕
予約お客様番号
予約者氏名
代理店コード
予約日時
1
*
*
1
1
*
*
1
*
1

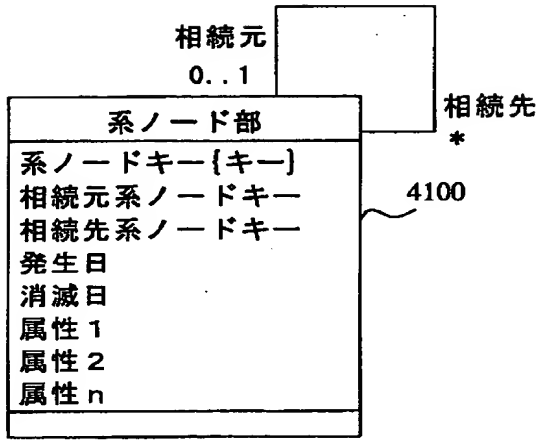
【図３９】


3900
機会部
機会キー〔キー〕
資源グループキー
3910
資源グループ部
資源グループキー〔キー〕
3930
オカーレンズ部
機会キー〔キー〕
資源キー〔キー〕
予約キー
3920
資源部
資源キー〔キー〕
資源グループキー
3940
予約部
予約キー〔キー〕

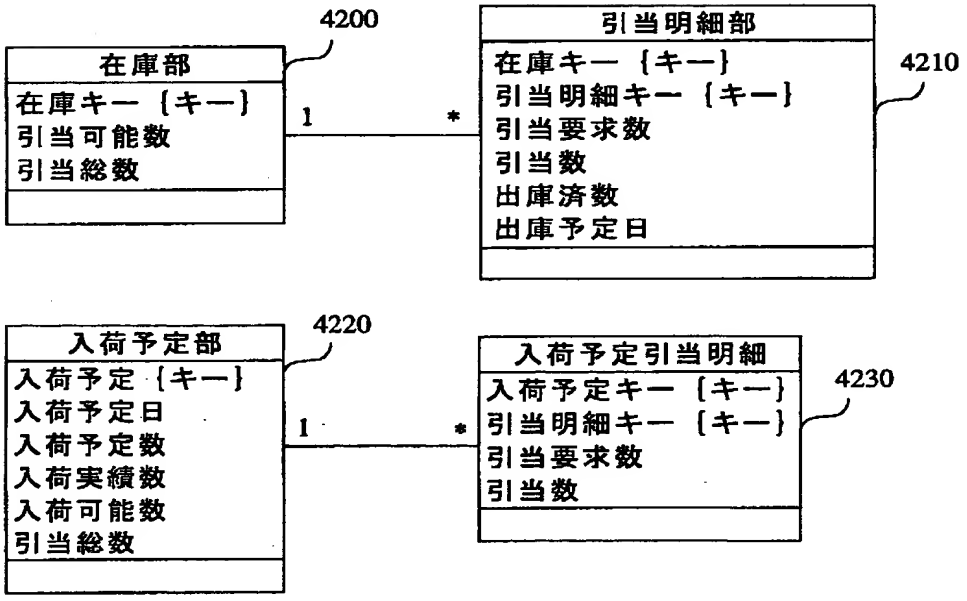
【図４０】


新規
1
4
新規
2
新規
3
消滅
凡例
：相続元
：相続先

【図４１】


相続元
0..1
相続先
*
系ノード部
系ノードキー{キー}
相続元系ノードキー
相続先系ノードキー
発生日
消滅日
属性１
属性２
属性ｎ
4100

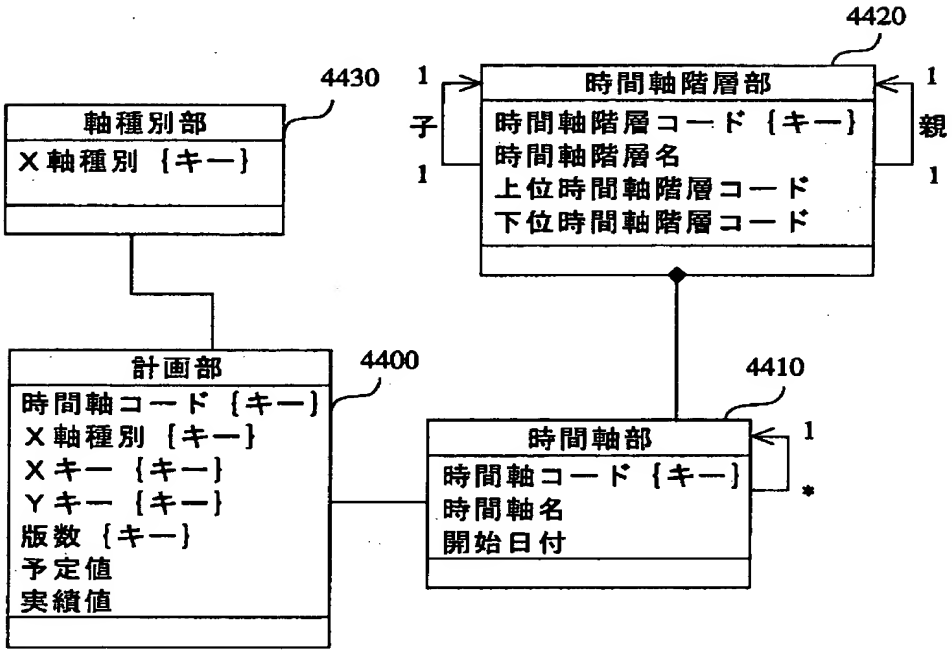
【図４２】


4200
在庫部
在庫キー {キー}
引当可能数
引当総数
1
*
引当明細部
在庫キー {キー}
引当明細キー {キー}
引当要求数
引当数
出庫済数
出庫予定日
4210
4220
入荷予定部
入荷予定 {キー}
入荷予定日
入荷予定数
入荷実績数
入荷可能数
引当総数
1
*
入荷予定引当明細
入荷予定キー {キー}
引当明細キー {キー}
引当要求数
引当数
4230

【図４３】

製品（X軸）	組織（Y軸）	時間(T軸)		
		年度	月度	日次
製品種	本部	○	—	—
	事業部	○	○	—
	部	—	—	—
機種	本部	—	—	—
	事業部	○	○	—
	部	○	○	—
製品	本部	—	—	—
	事業部	—	○	—
	部	○	○	○

【図４４】



【書類名】　　要約書

【要約】

【課題】

より少ない雛型を用いて、多様なプログラムを生成できるようにするプログラム自動生成技術を提供すること。

【解決手段】

所定の処理を行うプログラムを自動的に生成する本発明のプログラム自動生成装置は、各々所定の処理固有の設定を行うための解決ロジックを含み且つ予め対応付けられたデータ構造のための雛型プログラムを含む、複数のデータ構造解決ユニットと、選択されたデータ構造に対応するデータ構造解決ユニット内の雛型プログラムに含まれる解決ロジックの、所定の処理固有の設定に関する解決情報を取得し、当該解決ロジックの解決情報と雛型プログラムとを合成することにより、所定の処理を行うためのプログラムを生成する解決器とを有する。本発明ではデータ構造に対応して雛型プログラムが用意される。用途／処理フロー毎に雛型を用意するよりは、より少ない雛型の数にて多様なプログラムを生成できるようになる。

【選択図】　図１

出　願　人　履　歴　情　報

識別番号　　〔０００００５２２３〕

１．変更年月日　　１９９６年　３月２６日
［変更理由］　住所変更
住　所　　神奈川県川崎市中原区上小田中４丁目１番１号
氏　名　　富士通株式会社